TURING 图灵程序设计丛书

MANNING

Location-Aware Applications

LBS应用开发



[西] Richard Ferraro
[土] Murat Aktihanoglu 著

李丽 译

- 零基础掌握位置感知应用
- LBS应用专家创业真经
- 技术与商业并重的入门佳作



人民邮电出版社
POSTS & TELECOM PRESS

数字版权声明

图灵程序设计丛书

Location-Aware Applications

LBS应用开发



[西] Richard Ferraro
[士] Murat Aktihanoglu 著
李丽 译

人民邮电出版社
北京

图书在版编目（CIP）数据

LBS应用开发 / (西) 费拉罗 (Ferraro,R.) , (土) 阿克蒂哈诺格卢 (Aktihanoglu,M.) 著 ; 李丽译. -- 北京 : 人民邮电出版社, 2012.11
(图灵程序设计丛书)
书名原文: Location-Aware Applications
ISBN 978-7-115-29534-7

Ⅰ. ①L… Ⅱ. ①费… ②阿… ③李… Ⅲ. ①移动终端－应用程序－程序设计 Ⅳ. ①TN929.53

中国版本图书馆CIP数据核字(2012)第235921号

内 容 提 要

本书分为三部分，第一部分主要从宏观角度介绍位置感知及 LBS 应用，第二部分介绍与开发 LBS 应用相关的技术，第三部分讨论基于 LBS 应用的商业和创业。从介绍 LBS 的概念入手，本书涵盖了定位技术、地图选择、应用类型、移动平台、连接问题、服务器端整合，还为创业者提供了商业建议，涉及如何创收、保护隐私、发布应用、保护知识产权和寻求风险投资等内容。

本书适合移动应用设计开发人员和有志在移动大潮中成功创业的各领域人士阅读。

图灵程序设计丛书

LBS应用开发

◆ 著　[西] Richard Ferraro　[土] Murat Aktihanoglu
　译　李　丽
　责任编辑　李松峰

◆ 人民邮电出版社出版发行　北京市崇文区夕照寺街14号
　邮编　100061　电子邮件　315@ptpress.com.cn
　网址　http://www.ptpress.com.cn
　北京　印刷

◆ 开本：800×1000　1/16
　印张：16.5
　字数：389千字　2012年 11 月第 1 版
　印数：1－3 500册　2012年 11 月北京第 1 次印刷

著作权合同登记号　图字：01-2011-5232号

ISBN 978-7-115-29534-7

定价：59.00元

读者服务热线：(010)51095186转604　印装质量热线：(010)67129223

反盗版热线：(010)67171154

版 权 声 明

前　　言

从很多方面来说，2010 年都是 LBS（Location-Based Service,基于位置的服务）的一年。从技术角度和市场角度来看，这一年发生了很多 LBS 发展的里程碑事件。从技术层面来说，具有 GPS 功能的手机最终从一个狭义的新市场成为一个新领域。这种变化归功于 GPS 芯片成本的降低以及以诺基亚为首的移动市场的领导者的驱动作用。从市场定位来看，2010 年 LBS 应用的数量急剧增加。很多创业公司和老牌企业都宣称要在这个激烈的竞争市场中分一杯羹。现在，LBS 从一个独立的领域转而成为移动应用系统内的一个重要组成部分。位置响应应用逐渐成为移动应用的标准配置，尤其在智能手机方面。

真正的转折点是 2008 年年中由美国 Loopt 公司发起的首轮 LBS 电视广告运动，该公司在 Black20.com 上赞助了 the Middle Show[①]。iPhone 和 iPad 取得了巨大的成功，这使使用基于位置的应用变得简单而有趣，并成为当今移动生态系统的一个关键新领域。已经有逾 20%的 iPhone 应用以某种方式内置了位置服务，并且这个比例还在上升。Android 的发展进一步成为 LBS 应用的催化剂，刺激部署和采用更多的内置 LBS。

在这本书里，我们将说明通常哪些问题是位置应用开发者要关心的核心问题，而哪些适用于更广泛的移动生态系统：

- 怎样为你的应用选择正确的移动平台？
- 怎样从你的应用中赚钱？
- 怎样使你的应用脱颖而出？
- 怎样处理隐私问题？

我们希望本书能提供这些问题的答案和更多的信息。欢迎你来到“位置无所不在”的时代。

① The Middle Show 是 Black20.com 推出的网络系列剧。——编者注

致　　谢

这本书显然是大家共同努力的结果，在此表示感谢!

首先，感谢 Manning 出版社，他们的专业素质和奉献精神让人折服。感谢 Marjan Bace、Troy Mott、Cynthia Kane、Linda Recktenwald、Katie Tennant、Susan Glinert Stevens、Janet Vail、Mary Piergies 以及很多幕后工作人员。

感谢以下审稿专家，他们提出的宝贵意见和建议，使这本书更加完善：Todd Sabuncu、Darren Neimke、Amos Bannister、Michael Brennan、Jay Blanchard、Valentin Crettaz、Nikolaos Kaintantzis、Jeroen Benckhuijsen、Tyson Maxwell、Dr. Florian Resatsch、Massimo Perga、Gabor Paller、Orhan Alkan、Horaci Macias、Justin Tyler Wile、Curtis Miller、Jeff Addison、Eric Swanson 以及 Mala Gupta 。

特别感谢 Ben Allen，感谢他在收尾阶段全面的技术审核工作。

最后，我们还要感谢 LBS 社区，是社区一直以来的支持与反馈，以及他们对这个领域的极大关注成就了本书。

Richard Ferraro

作为移动创业公司 GeoMe 的共同创始人之一，那富有激情的两年工作经历很大程度上成就了本书。由于在这个领域有一些领先的想法，我花费了相当多的时间与这个行业的一些其他初创公司、技术供应商以及风险投资公司进行沟通交流，还参加了欧洲、亚洲的许多研讨会。我发现，在销售产品之前，必须对这个市场进行培训。这个过程赋予了我对 LBS 高超的洞察力和真正的感知能力，否则这本书就不可能完成。特别感谢移动开发公司 Golden Gekko 的现任 CEO Magnus Jern，他从始至终都充分信任我，给了我领导 GeoMe 项目这个绝佳的机会。我还要感谢 Manning 出版社的 Troy Mott，是他最先找到我，提出出版本书的想法，并给我写作的信心。最后，特别感谢我的父母 Denise 和 Franco、兄弟 Luca，感谢他们容忍我近两年不能陪在他们身边，让我周末和假期专心写作。

Murat Aktihanoglu

感谢我的父母 Orhan 和 Gunse，感谢我的妻子 Yoshiko，同时感谢我事业生涯中的良师益友 Rob Myers、Daniel Woods 和 Albert Wenger，以及我的同事 Chris Marrin 和 David Frerichs。

关于本书

尽管 LBS 以某种形式已经存在了近十年，由于技术的复杂性和快速发展，人们很难理解这些服务的原理。本书的首要目标就是帮助读者突破障碍，提供从定位方法到地图选择、从移动开发平台到内容图层等 LBS 有关元素清晰而简洁的解释。

LBS 将会让移动产业面貌焕然一新，塑造当前所谓的 Mobile 2.0 或多人参与的 Mobile Web 环境（相当于 Web 2.0）的核心。本书的第二个目标就是为那些初次接触 LBS 的开发人员或新的 Web 程序员提供该领域的一些真知灼见，激励他们利用 LBS 开发出下一个杀手级应用。我们会向你介绍典型 LBS 服务的构建，与你分享经验教训，这样你就可以避开很多常见的陷阱（请留意书中的提示部分）。

本书的第三个也是最后一个目标是阐明这样一个事实，即我们意识到越来越多的应用程序开发者同时也是创业者。他们渴望将开发热情与创办企业结合起来。对于大多数新近推出的网络或移动服务，货币化在初创企业的成功方面发挥了相当重要的作用。为此，除了介绍如何使你的应用从竞争激烈的移动应用中脱颖而出之外，我们还会讨论如何融资和经营。

目标读者

本书的核心内容是用通俗的语言讲解的，连门外汉都能看得懂，本书主旨是让移动世界的新人快速了解成功的 LBS 所需的开发、构建以及发行的知识。本书不是工程手册或者深奥的位置技术理论专著——如果你想了解方位制表和接收器定标的话，建议你选择其他读物。本书会在特定的地方详细探讨程序开发细节和代码，只是为了给移动应用开发人员提供一个实用工具，好启动 LBS 项目或在他们现有的项目中添加特定的功能。

如果你符合下面这些情况，那么这本书很适合你：

- 你是一名 Web 程序员，希望学到更多开发移动 LBS 应用的知识；
- 你是移动行业的高级管理人员（CIO 或者 CTO），希望了解移动开发这个充满活力的领域；
- 你是一名企业家，希望探索如何利用 LBS 赚钱；
- 你是一名刚刚接触 LBS 开发的移动程序员，希望了解所有与 LBS 相关的知识；
- 你对移动互联网有着强烈的兴趣（也许你也订阅 *Wired* 杂志），希望知道 LBS 将如何革新移动行业。

组织结构

本书的特色在于，它既是一本技术图书，也是一本商业图书。

在这本书中，我们以技术手册的风格融汇了两种体裁。这样在开始时你会了解 LBS 的概况，到了中间就能知道开发中的疑难问题，最后明白移动商务的战略。

为了让本书更轻松易读，也为了符合此书最初编写的目标，我们将它分为三个部分。

第一部分，介绍 LBS 概况以及使位置感知应用和服务广泛采用的技术。

第二部分，全面概述如何将 LBS 付诸移动应用实践：选择什么样的平台及其原因，怎样在此基础上成功构建。我们在这一部分专门增加了一章来介绍当前的消费者应用，好让读者将实用工具和利用这些工具开发的应用联系起来。

第三部分，本书最后一部分，讨论 LBS 移动应用开发的商务方面，它能使你的研发成果更有可能被广泛采用，更容易成功、更有利可图。

你也可以根据自己的兴趣挑选喜欢的章节开始阅读。不管你偏爱什么，至少要看完第 1 章、第 3 章、第 4 章和第 5 章，因为这几章能使你对 LBS 的理论和实践有个大体的了解。

编码约定和下载

本书有不同编程语言的代码实例，比如 Java、Objective-C、PHP、HTML 和 JavaScript。我们会采用每种语言的习惯做法，这样易于你亲自试验这些代码实例。

通常我们采用以下语法规则：

- 包名称都是小写字母；
- 类名称首字母大写，中间的单词首字母大写；
- 变量首字母小写，中间的单词首字母大写；
- 每行最多有一条语句。

所有代码清单中的源代码都采用等宽字体，与普通文本区分开。很多清单中都有代码注释，用以强调重要概念。有些情况下，代码清单中会有数字编号，这些编号与代码清单后面的相关解释中的编号对应。

本书中的代码示例可以从出版商网站www.manning.com/ferraro上下载[1]。

作者在线

购买本书你就可以免费访问 Manning 出版社的内部论坛，你可以在这里发表关于本书的评论、询问技术问题以及从作者和其他用户那儿获得帮助。你可以在 www.manning.com/ferraro 上访问或订阅该论坛。这一页提供注册后登录论坛的信息，可获取的帮助信息以及论坛规则。

① 读者也可以登录图灵社区本书页面（ituring.com.cn/book/865）下载。——编者注

Manning会对读者恪守承诺，为读者之间以及读者和作者之间提供交流途径，但并不承诺作者的参与度，他们的参与本着自愿的原则（无报酬）。我们提议你多向作者提一些挑战性的问题，好引起他们的兴趣。

本书一经出版，Author Online论坛以及以前的讨论资料都可以通过出版商的网站访问。

关于封面图片

本书封面上的画像标题为“护林员”，摘自 19 世纪法国出版的沙利文 · 马雷夏尔（Sylvain Maréchal）四卷本的地域服饰风俗纲要。其中每幅插图都是手工精心绘制并上色的。马雷夏尔这套书展示的丰富服饰，令我们强烈感受到 200 年前的乡村及城镇的巨大文化差异。不同地域的人山水阻隔，言语不通。无论奔走于街巷，还是驻足于乡间，通过他们的服饰，一眼就能看出他们的生活场所、职业，以及生活境况。

时过境迁，书中描绘的那些区域性服饰差异到如今已经不复存在。即使是不同国家，都很难再看出人们着装的区别，再不必说市镇和乡村了。或许，我们今天多姿多彩的人生，正是从前那些文化差异的体现。只不过，如今的生活更加多元，而且技术环境下的生活节奏也要快得多。

今时今日，计算机图书层出不穷，Manning 就以马雷夏尔这套书中多样性的图片，来表达对 IT 行业日新月异的发明与创造的赞美。

目　录

第一部分　LBS 蓝图

Part 1

第一部分

LBS 蓝图

本书第一部分主要从宏观角度介绍位置感知及 LBS 应用。

第 1 章将讨论什么是 LBS，以及构成 LBS 的不同组件。此外，还会从不同行业及全球的视角来俯看 LBS 应用。

第 2 章结合实际讨论今天的定位技术，重点介绍各种定位技术在构建位置感知应用时的优缺点。

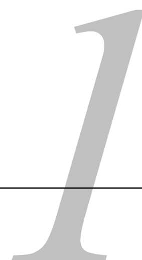
第 3 章则讨论开发应用时可选的地图，也会涉及目前市面上的开源地图程序。

尽管市场上有数以百万计的应用，但内容仍然是核心，是区分其他竞争者的关键。本部分最后一章，第 4 章，将介绍应用将包含哪些内容，包括地图中的分层和混搭。

读完本书这一部分后，你会了解移动设备上的 LBS 如何运作，以及这个技术为开发者和企业提供了多么美好的前景。

第 1 章

基于位置的服务



本章内容

- 基于位置服务的要素
- 商务 LBS 和用户 LBS
- LBS 移动开发的挑战
- LBS 未来的机遇

位置，位置，位置!

“买房时最重要的问题是什么?”对这个常见问题的回答，很快就同样可以回答以下问题:“当前移动服务中最重要的问题是什么?”

为什么位置如此重要呢?

各种新奇、令人兴奋的移动网络服务给我们的日常生活带来了翻天覆地的变化，而位置则是这场变革中的基本要素。当今，最流行的移动应用和服务的有效性取决于一个关键因素：当你正在使用该项服务时身在何处。不论你是登录社交网络还是在午夜寻找药店，关键要素通常是一样的：位置。位置服务最早只是手机中一个吸引人的附加功能，而现在知道自己的位置以及怎么到达某个地方已经成为人们日常生活的基本需求。

本书将探索无限精彩的 LBS（Location Based Service，基于位置的服务）以及位置感知应用。它们最初只是移动运营商招徕顾客的一种手段，但是越来越多的消费者希望能随时随地使用这项服务。谈到基于位置的服务，我们所指的应用和服务非常广泛，它们旨在高效运行在移动设备上，而且都利用了某种形式的定位或“基于位置”的技术。

这一章的目的就是介绍迷人的 LBS 世界，以及它是如何快速发展为全球移动应用中最重要的一个门类的。除了要介绍现有的各种 LBS 服务，我们还将展望一下它未来的美好发展前景。但是在讨论这些令人兴奋的内容之前，首先需要再多了解一下基于位置的服务的含义。

1.1 什么是基于位置的服务

首先，让我们从基础知识开始。每个人都知道纸质地图，越来越多的人知道在移动设备上可以使用电子地图。这些最初简单的地图在添加了一些数据层或者兴趣点（Point of Interest）

之后，可以让使用者看到商店、加油站等的位置。这样一来，这些地图就大致成为了一种简单基于位置的服务。

通常情况下，可以将基于位置的服务简单定义为：通过移动网络连接、可由移动设备访问，同时利用移动设备地理位置的一种信息服务。

这个定义在当前的移动和网络（所谓的 Web 2.0）时代已经过时了。现在，用户参与（是的，那就是指你！）才是最最重要的，而产生内容的能力是这些服务和应用的核心。

今天，对 LBS 更恰当的定义应该是，通过它：

- 用户能够确定自己的位置；
- 可以看到与用户位置在空间上相关的信息；
- 用户能够与相应的位置信息或内容动态或者双向地互动。

这样的话，用户可以回答三个关键问题：

- "我在哪里？"
- "我在附近能做什么？"
- "我对这个地方看法如何？"

现在，我们已经知道 LBS 的最新定义了，接下来看看 LBS 的组成部分。

LBS的关键组成部分

基本概念介绍完了，现在让我们深入了解一下 LBS 的组成部分。所有 LBS 应用系统都有四个组成部分：移动设备、内容提供商、通信网络以及定位组件。如图 1-1 所示，这四个部分形成了 LBS 的价值链。

现在，我们分别详细讨论一下每个组成部分，看看它们各自都能给 LBS 带来什么价值。

1. 移动设备

移动设备的多样性使 LBS 应用的开发变得更加复杂，必须恰当考虑这一因素的影响，特别是服务上线之前的测试阶段。

我们所说的移动设备，是指能够（通过 SIM 卡）连接移动网络并传递语音和数据（比如打电话和下载电子地图）的电子设备。

对移动应用开发者来说最关键的问题是，基于 PC 的 LBS 服务只能让用户在自己家里使用，而 PC 不能随身携带、想用就用。这种情况限制了 LBS 服务的有效性，同时只能局限于一个很小的用户群。而移动设备的出现大大扩展了 LBS 的市场，使其成为一种炙手可热的户外应用形式。

虽然 Symbian OS 在众多移动平台中比较突出，约有一半的手机使用这个操作系统，但是智能手机赢得越来越多用户的喜爱，这种情况导致越来越多的手机采用其他的操作系统，比如 RIM、Windows Phone7、Linux、ios、Android 以及 Palm Web OS 等。

智能手机具有较大尺寸的显示屏幕（看地图更容易），几乎普遍内置定位技术（比如全球定位系统 GPS），对 LBS 的快速发展具有至关重要的作用。

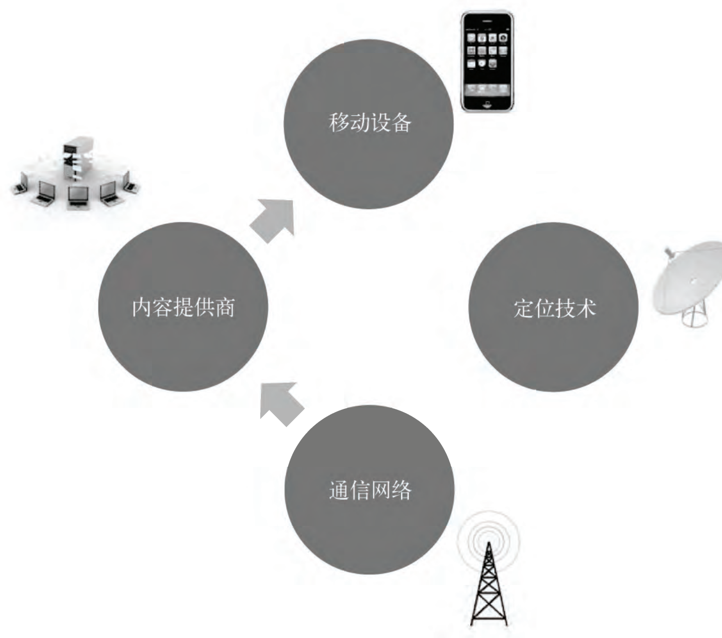


图1-1　LBS构成图展示了向用户提供服务必需的4个组成部分

最近，平板电脑可谓突飞猛进（受 2010 年苹果公司大肆宣传推出 iPad 的刺激，如图 1-2 所示），这就意味着开发者考虑的移动设备不应该仅仅是手机那么简单了。

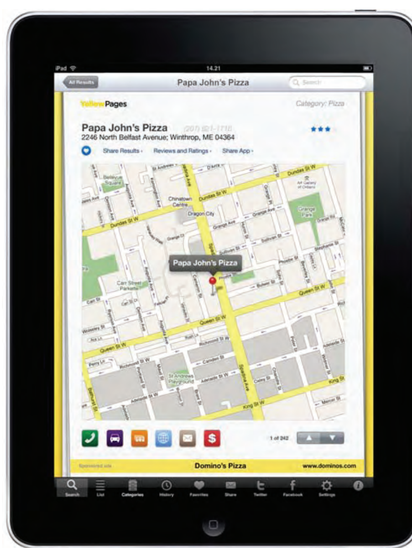


图1-2　这个iPad正在运行的LBS应用是Yellow Page（黄页）

连网移动设备上的位置服务之所以让人如此兴奋，主要是因为这种设备是移动的，而不仅仅是提供静态位置的信息。消费者只要不断刷新位置应用或服务所使用的内容，就能一键获取大量的信息。接下来，我们简略地介绍这些内容可能的来源。

2. 内容供应商

移动内容供应商是能够创造或者本身就拥有媒体（内容）的实体，它能把内容直接或通过第三方间接地提供给移动设备。随着为移动用户提供的移动应用和移动服务数量的激增，内容供应商的作用就变得越发重要了。这种情况促使很多应用开发商和经销商竭尽全力去开发更具吸引力的内容。

通常情况下，对于那些用户通过移动设备就可以访问的内容和数据，LBS 供应商并不会全部储存或保留。比较明显的例子就是地图数据，常常由 NAVTEQ 公司这样的地图供应商提供。渐渐地，用户可以通过第三方内容供应商提供的地图层来访问数据，并可根据自己的需求，选择打开或关闭这些数据（比如在地图上只显示出加油站的位置而不用显示药店的位置）。

内容，将越来越成为区分公司与竞争对手的关键因素，“内容为王”也是移动领域的法则。第 4 章将会介绍这方面的内容，了解一下与其他 Web 服务混搭（mashup）可以获得什么内容。

3. 通信网络

从另一个方面来讲，通信网络不是位置服务开发者所能直接掌控的，但是它却能在使用 LBS 服务管理数据传输时发挥作用（最大化移动速度或者最小化延迟时间，以及向按需付费用户限制数据流量）。通信网络也是 LBS 服务在互联网上首次展示时需要考虑的一个关键要素（例如要在最新的 3G 网络上实现地区性部署）。

现在我们继续看一看 LBS 的最后一个要素：定位组件。它关系到 LBS 中的定位，因此至关重要。

4. 定位组件

我们所说的定位组件，通常是指肉眼看不到的那个装置。也就是说，它内置于移动设备之中，追踪该设备的具体位置，然后向移动设备上运行的应用程序传递位置信息（比如，浏览地图时，地图上就会有某个点显示该设备所在的确切位置）。

对所有的 LBS 来说，这个装置的定位功能和有效确定自身所在位置的能力显然是必不可少的。我们会在第 2 章中对其做详细介绍。

目前，定位组件成为很多手机的一种内置预设功能。即使对应用开发者而言，LBS 的定位组件也变得越来越“隐身”。不过仍有不同的定位方法可供选择，包括三角测量、小区识别码（Cell ID）、卫星导航以及无线定位系统（WPS）。许多手机可以采用不止一种定位技术方法，而由于混合技术能够减少由单一定位技术带来的不利因素（我们将会在本章的后面进行介绍），其使用正变得日益频繁。

另外，至少能确定一个大致位置的 API（应用编程接口）或软件部件定位应用变得越来越普遍。移动浏览器对这种定位的关注持续增长，例如，对在本地的移动设备，可以在移动网站上给出搜索结果。

我们已经知道了 LBS 的几个关键组成部分，现在回顾一下 LBS 的发展历程。

1.2　商业 LBS 和用户 LBS

在过去的五年里，LBS 向全球手机用户提供的服务数量和服务种类急剧增加。LBS 大众市场的第一代产品，能利用无处不在的手机提供应急响应服务。后续的 LBS 服务从提供实用信息（比如最近的加油站在哪儿）转为更关注休闲活动。

如今，很多有关用户服务的 LBS 革新常包含社交网络或社区元素。在某种程度上，这是由于基于 Web 的社交网络迅速升温，而这一现象必将会漫延到移动世界。

事实上，利用手机定位功能的商业或者 B2B 服务是目前的 B2C 应用模式的前身，而且已经建立了切实可行且可持续发展的商业模式。

这一节我们会了解一下在 LBS 应用较成熟的行业里它的部署情况，比如运输业（包括无线电频率识别如何进一步提高 LBS 应用的有效性）。看一看目前 LBS 新的服务项目所取得的成就，这些新的 LBS 服务正在蓬勃发展的休闲领域努力开拓、寻求新的发展。

1.2.1　GPS在交通运输业

2005 年的交通运输和物流营业额超过了 3.5 万亿美元，在全球贸易快速增长的大背景下，货物运输仍将是一块大蛋糕。难怪会将手机作为昂贵的物流及跟踪系统的廉价替代技术，并进行猛烈的宣传！

手机已无处不在并且内置了位置跟踪功能，人们为什么还要花费巨资定制 IT 基础设施，购买软件和硬件呢？这种现象确实存在，尤其是在新兴国家。在那里，计算机、IT 和电信基础设施还不发达或者太过昂贵。

比如在泰国和马来西亚，有一个叫 MappointAsia 的公司，通过手机和 GSM 移动网络获取 GPS 定位信息，向用户提供人员和车辆跟踪、车队管理以及寻找被偷车辆这种低收费的服务。

美国 TeleNav 和以色列 Telmap 等公司抓住了机会，利用日益成熟的移动设备上较为完善的定位技术，为自己的导航软件开辟了一个新市场。两个公司都可以提供劳动力管理、车辆管理工具和车载导航服务。我们将会在第 5 章里详细介绍这些公司。

1.2.2　GPS与RFID对比

无线射频识别（RFID，Radio Frequency Identification）是一种无线电技术标准，定义如何利用无线电波传送某物或某人身份（一种唯一的序列号形式）。按自动识别技术[1]RFID 可以分为很多类。

RFID 系统由传送装置（指标签）和接收装置（指读取器）组成。标签相对简单，可以向另外一台装置发送信息。RFID 的可读距离从几厘米（被动标签）到几百米（主动标签）不等。小型化的 RFID 芯片可以小到 0.05 毫米，这个长度的芯片可以植入人体内部了（能植入人体的 RFID

① 参见 www.rfid.org。

芯片如图 1-3 所示）。



图1-3　RFID芯片的形状和尺寸多种多样，比如可植入芯片，可以用于医疗保健行业诊断疾病

RFID 通常用在物流或者高速路自动付款上，其优于 GPS 的关键方面是，电子标签既不需要电源也不需要瞄准线。

内置或者安置在手机上的 RFID 芯片可以发送数据（比如日本的 NTT DoCoMo 公司就开发了一种允许用户把手机当作门钥匙的技术），但不能用于检测手机的绝对位置。

在将来的商业 LBS 应用中，安装了 RFID 芯片的手机很有可能成为混合位置服务的一部分，这样的手机能够用于在各种情况下定位移动资产。这样，就可以用 RFID 元器件追踪资产，用 GPS 定位并将其位置显示在手机的电子地图而不是独立（且价格更昂贵）的跟踪装置上。

1.2.3　紧急响应服务

用手机能够获悉紧急情况发生的确切位置，这种能力使其在紧急援助和救助服务中成为不可或缺的工具。比如儿童走失、老人独自一人或者一个大学生遭遇绑架的情况。

美国的 Rave Mobile Safety①这样的公司生产的手机有报警功能。大学生可以将手机当作报警器。谁觉得有危险时就可以激活手机上的定时器功能，如果过了设定的时间定时器还开着，它就会拉响校园的安全警报。一旦情况发生，就可以用 GPS 确定学生的位置。

与 Rave Mobile Safety 提供的手机报警定位功能相似，英国的 KidsOk 公司推出了儿童定位服务。不过这项服务提供额外的安全特性（包括本机加密功能和父母审查功能），以便保护儿童不被跟踪。

① 公司以前的名称是 Rave Wireless。——编者注

西班牙的路边援助服务 RACC 在 2009 年推出了 FindMe（找到我）服务。用户可以通过手机发送 SMS 获知朋友和家人（先要得到他们的同意）的位置信息。最初，这项服务只用在处理路边事故和突发事件上，现在推广到一般大众了。

1.2.4　服务定位软件的应用

过去几年间，LBS 服务从最初的核心应急响应服务演变为向人们的日常生活提供简单、实用的解决方案（通常涉及城区内的寻人及辅助设施）。

位于美国马萨诸塞州波士顿的 uLocate 曾站在实用 LBS 新浪潮的前沿。在其 WHERE(在哪里)平台（后来被 eBay 收购）上，它开发了各种各样针对不同用途的定位部件，可以通过美国各大无线运营商订阅使用。

比较有名的例子是 Zipcar 汽车共享部件。会员可以在手机上利用汽车共享部件定位查询最近的 Zipcars，也能立即联系 Zipcar 的预定系统，选择一辆就近的汽车。利用 NearBio 部件，用户可以快速找到离自己最近的提供生物柴油的加油站。

快速浏览 iPhone 手机应用中的苹果 iTunes 商店，你就会发现这样的应用增长已经相当迅速了（在 iTunes 商店里，超过三分之一的应用实际上都是有关位置感知的。这些应用能够提供很多的实用信息，比如 Urbanspoon 的饭店搜寻器）。

MapQuest Maps 和 Ovi Maps（以前是 Nokia Maps）都有着丰富的兴趣点（POI，Point of Interest）和地图图层数据。从当地的公共交通到药店，这些地图可以提供全面的实用信息。在第 3 章里我们将会详细介绍地图选项和地图叠加的相关细节。

1.2.5　社交网络应用

我们生活的世界，媒体消费过度，科技设备（包括手机）无处不在，但是人们却很容易厌倦。越来越多的手机用户叫嚷着需要更多的手机功能来休闲和娱乐。

想想看，最成功的 iPhone 应用中居然有些一次性的古怪玩具，比如“猎枪”（不用说你也知道，可以发出猎枪开火的响声）和“打火机”（对，可以看到火焰）之类的，还能有其他原因吗？

今天的青少年被称作是网络的一代。他们伴随着网络长大。那个虚拟的数字世界是他们身份的逻辑延伸。网络中的数字身份和现实中的真实身份对他们而言同等重要。他们希望手机（毕竟，是他们关键的身份象征）能够访问网络中的数字身份是很自然的事情。

移动社交网络向这些网络一代所提供的正是他们所需要的——随时随地与朋友保持联络。全球基于 Web 的社交网络巨头，美国的 Facebook 和 MySpace，日本的 mixi 以及韩国的 Cyworld 拥有超过 6 亿的用户。

2007 年，移动世界的第一批社交网络（比如 Loopt）出现。这些社交网络通常与手机搜索出的位置信息相连。从 2008、2009 年开始，“开放式”网络模式成为主流。很多主导社交网络（以 Facebook 的 Connect 服务为首）的 API 对外公开。好了，基于位置的社交网络用户突然发

现，他们所能体验的服务更加丰富了。作为第三方的社交网络可以通过一个巨大的社交图谱[1]联系起来。

从那时起，基于位置的移动社交网络数量开始激增。这种网络强调趣味性。前所未有地，其规模遍及全球，并快速成为 LBS 供应商和应用开发商的创新阵地和商业战场。

图 1-4 显示了全球可用的这类服务中的一些，既有刚创建的小型公司，也有一些成熟完善的公司。韩国和日本这样发达的移动市场，分别在 1999 年和 2004 年出现了 Cyworld 和 mixi，总用户超过了 3400 万。



图1-4　流行于世界各地、基于位置的移动社交网络分布图。这些网络日益流行，估计其用户总数超过6000万

有趣的是，像前面提到的 TeleNav 和 Telmap 这样的最初专注于实用 LBS 或商业 LBS 的公司，针对用户的需求，正试着在其服务项目中增添社交网络类的服务。这种现象将来很可能会造成一定程度的趋同，造成服务供应商向终端用户提供集成化服务的现象，其原因就在于不同类型服务供应商的边界变得越发模糊了。

我们已经粗略了解了当下不同领域中 LBS 有趣的服务与创新，有必要看一看 LBS 在发展中遇到的挑战，这样更有助于扩大 LBS 的服务范围，增强 LBS 的服务吸引力。

1.3　移动 LBS 在发展中遇到的挑战

面向大众推出有效且全球可用（意味着移动电话运营商、国家与手机制造商互不依赖）的 LBS 非常有挑战性。如果不具有挑战性，也许在移动 LBS 领域，移动开发商的前景就不会那么诱人了。

[1] 社交图谱（social graph），是由 Face book 在 2008 年提出的概念，指一个将人们和不同事物联系起来的地图。——编者注

采用 LBS 应用需要考虑三个主要挑战：

- 程序通过无线网络访问数据的成本；
- 手机厂商和运营商；
- 隐私法规和公众认知。

1.3.1　访问成本

移动应用开发商和一些此领域内刚创立的公司通常都想追随某一利基或市场细分。这个市场有可能与平台相关，比如专门为 iPhone 开发的服务，或针对某个地区专门开发的服务，例如只能在北美地区使用。

实际上，由于手机在异地有漫游费，相较于很多其他的移动服务甚至网络服务，地域市场的选择是更关键的业务决策。

确实，移动应用大众市场的开发人员仍需面对一个关键问题，那就是克服数据使用（获取位置的固定信息或是反馈与位置相关的动态信息，这些应用都会用到数据）过程中产生的（经常过分高昂的）漫游费用。在一个国家内，漫游费用或许不能算问题；但是对于那些计划用在很多国家的应用（像旅行应用程序）而言，漫游费用是一个很大的问题（尤其在欧洲大陆，有时国与国之间的边界只有几百公里）。如果打算在用户居住国之外使用该应用的话，就会产生所谓的“手机账单休克”这样实际的问题。成熟的市场中很可能会普遍对移动数据访问按固定费率（或者说“任意享用”方式）收费，应用开发人员仍有义务对按字节收费的终端用户减少数据传输量。

即使数据漫游费用不再是问题，也必须将用户访问网络应用产生的费用设定在一个合适的水平上，这样才能鼓励大众采用。传统订阅模式最初在定位服务应用中运行良好，消费者乐于每月花 3 美元选择这些应用。对于社交应用而言，采用由广告收入支撑的免费模式要多于采用某种标准收费的模式。我们将会在第 8 章了解一下能够在 LBS 应用中植入的各种广告，也会看看用于 LBS 收费的各种定价模型。

1.3.2　手机厂商和运营商

一谈到手机厂商，必须知道不同厂商都有各自不同的约束条件。有些可以提供更可靠的 GPS 定位信息，寻找特定的厂商是非常必要的，所以要创建一个应用项目，这些约束条件就是选择厂商的关键。

另外一些厂商可以提供开放的、允许用户访问的开发者社区（Nokia Forum 就是一个很好的例子），这样用户就可以在一个开放的大论坛上讨论一些日常发现的问题了。

还必须考虑高端手机以及具备定位功能的智能手机有多普及。如果要面向大众市场，那么在最广泛可用的平台上开发就是关键所在。而像 iPhone 这样特定的智能手机可能更适合于开发支付额外费用以及其他方面的应用。随着科技不断发展，移动平台相互混合，在正确的时间发布正确的产品是很必要的。虽然早在 2008 年，大部分 Android 平台的发展都处在试验阶段，但到现在这个平台的发展势头如此之好，以致于多数手机开发人员不得不转向 Android 平台开发。

实际上，谷歌为了占有更多的移动市场份额，在 2010 年就开始发力，他们推出了 NexusOne（如图 1-5 所示），尝试将手机运营商与移动应用分离开来。他们尝试绕开中间的手机运营商而直接向用户销售手机。这就使手机运营商距离纯数据载体（或“哑管道”，就像网络服务供应商一样）又近了一步。后来，谷歌停止了 NexusOne 手机的直接销售（2010 年 7 月），终结了这种模式的所有销售工作。与其说是这种直销模式的失败，不如说是他们手机营销的失败。

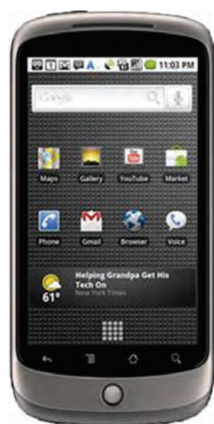


图1-5　2010年1月，谷歌公司推出NexusOne智能手机，采用最新版本的AndroidOS，这标志着谷歌试图获取移动市场更多控制权

所幸的是，移动运营商对移动应用开发所能施加的影响整体上正在弱化。以前，一款手机应用只有通过移动运营商的批准“入户”，才能确保得到有效地发行。今天，“户外”模式证明自己是行得通的，消费者用实际行动证明了他们更喜欢通过第三方应用经销商或者在应用商店购买自己喜欢的产品（不过后来移动运营商顺应潮流，开办了自己的应用商店）。我们会在第 11 章了解这些分销渠道。

1.3.3　隐私立法和公众认知

一些国家对公众隐私有非常严格的法律规定（法国就是其中一个）。因此，在运营商的网络上，手机用户所在的位置会受到多层授权保护。

在其他国家，民用的 GPS 定位要么违法（利比亚）要么受政府控制。

要想 LBS 在移动通信中应用更广泛、利润更可观，决定因素有很多。毋庸置疑，其中一个关键问题就是所谓的隐私难题。

消费者需要保护，在潜在风险中数据隐私位居前列。一旦允许对消费者定位，很简单，就相当于将他们的行为暴露在公众面前，很多人对此表示担忧（更不用说，通过手机信号，移动运营商在任何时候都会知道手机用户身在何处了！）。

消费者担心如果自己的位置能被查出，那么善意的谎言就有可能曝光（举个例子，你告诉同伴正去健身房，而实际上正与一群朋友喝酒）。一般更多的人则是担心被陌生人跟踪。这样矛

盾就出现了，一方面使用该服务的用户需要第三方访问其隐私信息，而同时又担心这些信息会被滥用。

对于用户隐私，运营商引进了一些比较明确的可变隐私设置，缓解了消费者的担心。比如：用户可以自行决定 LBS 查找他们位置的定位精度。一些观察家就此提出，在将来的某个时候，用户会由于位置响应服务“联系不上”，或者处于离线状态而付费。

消费者对数字服务侵犯隐私的强烈反对导致了隐私立法，其依托基础在美国、日本及欧洲均相同。隐私立法引进了一个关键因素：*准许*。比如欧盟 2002/58/EC 四号指令第九条声明，除了通信数据，可以对与公共通信网络或公开的电子通信服务相关的用户位置数据进行处理。但是要处理这些数据，必须在用户匿名状态下或者征得用户准许，在某种程度上、一定期限内，为用户提供增值服务的情况下才能处理。服务供应商必须在征得用户准许之前，对其告知除了通信数据之外需要处理的用户位置数据类型、处理目的、处理期限，以及这些数据是否会传送给第三方提供增值服务。除了通信数据之外，用户可以在任何时候，收回对其位置数据进行处理的许可。

这就意味着无论是在 LBS 应用开发，还是在服务的运营方面，关键就是要求其隐私政策中要有一个事先征得用户同意的条款。开发者必需牢记这一点。遵从这一规定的形式通常就是在应用程序里显示一个选择窗口。对于那些高级一点的应用（比如，使用“推送”机制的应用），移动运营商和移动设备制造商都有可能设置一些额外的条条框框，开发人员必须发布符合他们要求的应用。

我们将会在第 10 章详细介绍这些问题。

1.4　LBS 未来的机遇

众所周知，下一代的移动服务中位置正快速成为广泛普及并无所不在的一员。有确凿的证据表明位置服务将充满商机。

LBS 在 2009 年全球的订阅收入是 9.983 亿美元，但是到 2013 年预计将会达到 650 亿美元。而移动广告，还是一个朝阳产业，在 2009 年的全球收入就已经达到了 9.13 亿美元，预计 2013 年会达到 130 亿美元。有人猜想这项支出的主要部分将投向基于位置的应用。因此，再看到移动广告行业中的并购现象就不会觉得奇怪了。2009 年 11 月，谷歌花费 7.5 亿美元购买了移动广告平台 AdMob，而苹果公司在 2010 年 1 月耗资近 3 亿美元收购了 Quattro Wireless。与此同时，从不同应用商店下载的程序数量一直呈指数增长，由此产生的收入在 2010 年就达 62 亿美元，预计到 2013 年会达到 216 亿美元。

不是只有传统的商业模式会产生商机，在 LBS 领域有可能出现一种新的定价模式，这种定价模式也将促进当前领域之外的经济增长[①]。

移动世界是目前正在快速发展的科技领域，继传统 PC 销售的增长之后，手机的销售数量也在快速增加。随着新手机技术的进步，移动服务具有了新的功能（iPhone 手机上的加速计能够检

① “Mobile Content and Services,” 7th ed., Informa Telecoms & Media, March 2009。

测运动和速度就是一个例子），这就意味着移动应用开发正不断突破原有限制。

考虑到移动世界中这么多的革新，以及将这些革新推广到大众市场所固有的挑战，有必要看一下，在不久的将来，市场中到底会有哪些 LBS 产品。

到 2020 年，据说 LBS 将会提供无数的新功能，这些功能都是现在很难想象得到的。到那时，预计最新的移动网络基础设施（即 4G）将会在很多发达国家普及，其数据空中传输速度将会比现在快很多倍。功能更强大的手机将会有电脑般的处理能力，可以在各种环境下与机主以一种全新的方式互动。

多数互动有赖于内置于应用或设备中的智能工具。它能根据用户的使用环境，在某个合适的时间点自动提供相关的信息，推出相关的功能。

一切都要看环境

想象一下你正在山上滑雪，激活了手机上的定位功能。你想要在手机地图上显示天气预报。一个智能的 LBS 应用不仅会回馈天气预报，还会给你提供预期的雪情，让你知道明天是否能滑到粉雪。

之所以能做到这一点，是因为应用程序了解你所处的环境：能连接到网络（开着手机，在手机信号范围之内），能确定你的位置（在山里），另外知道当时的季节是冬季（环境），因此就能给你提供有用的滑雪信息。

图 1-6 显示基于环境服务的几个要素：在线、位置和环境。接下来举例说明 LBS 根据滑雪者的移动设备来获知用户需求的情况。

在线（presence）是指能联系到网络；*位置*（location）是指应用程序知道你在哪里；而*环境*（context）就是说应用程序知道你可能在做什么。

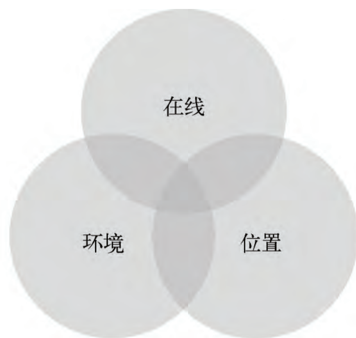


图1-6 以滑雪者为例来说明环境因素

这是一个关于应用程序能够实现完美上下文指示的基本例子。我们这样说是因为还没有一种应用服务能够利用用户所在的环境为其提供智能解决方案，推出目标信息。无论从哪个方面看，这都是应用程序开发者终极追寻的目标。谷歌开始了有关智能网络技术的研究，也许能够提供一种强大的机制来实现上下文感知（或*上下文搜索*，谷歌是如此定义的）。实现上下文感

知的基本原理就是根据用户的实时位置信息和其他因素，移动设备能够智能预料（或预测）用户正在寻找什么。谷歌公司的 Goggles（拍立寻）就是这样的初级尝试。用户用设备的相机来辨识物体，比如拿相机对准一个啤酒瓶，“拍立寻”就能提供其详细信息（比如配料、截止日期，等等）。

说到上下文，时间是一个基本参数。它能使应用程序更智能。能够感知其他变量的能力也正在研发之中。一些手机制造商已经推出了能够监控用户血压和心率，或监测空气污染物或花粉含量的移动电话。

下一代移动电话将为我们提供一个智能服务的全新世界。

比如，繁忙的一天工作之后，你马上要下班了，用手机（比如，2008 年诺基亚公司研发的一款，如图 1-7 所示）检测一下自己的心率。人有压力时，心率比正常值要高。



图1-7 诺基亚的一款生态传感器手机样机，它装有心率感应器，也能检测空气中的花粉含量，向过敏症患者提示环境危险

途经一个香薰店时，基于位置的环境响应程序就会提醒你，使用商店里的哪款产品能够降低心率。

这也许听起来像是讲述未来的电影《未来报告》（*Minority Report*），而实际上以现在的科技来说是能够联系环境的各个元素，从而为用户提供极致的上下文体验的。

由上下文感知，可以联系到环境感知[①]。环境感知大致相当于根据其他人在做的日常琐事来推断出他们在忙什么，或者相当于对某人的社交网络所从事行为的背景感知。

这个相对还比较新的概念在今天正快速的成为新型社交网络的典型特征。比较著名的例子就是 Twitter。Twitter 于 2006 年创立，是为了知道其团队成员正在做什么而频繁共享短信息的一种方式。目前，Twitter 的用户已经相当可观。

Twitter 可以提供用户的上下文信息，所以可以用在移动应用中替代直接从手机中获取的用户信息。实际上，结合 Twitter 内容与移动应用的这种模式（混搭应用）正快速成为移动社交网络的标准做法。自从 2009 年 11 月 Twitter 的 Geolocation API 推出以后，第三方应用软件现在就可

① 2004 年，荷兰埃森霍温技术大学工业设计部门的用户中心工程组提出环境感知（Ambient awareness）的概念。

以有地理标签信息或消息了，也能有照片了（如图 1-8 所示）。

这个领域的不断创新意味着，在各种各样的服务中添加定位功能的机会无穷无尽。



图1-8　2009年11月，Twitter发布了Geolocation API，能从用户微博中直接获取其地理标签内容，揭开了位置感知混搭应用的全新一页

2009 年 12 月 Twitter 的创始人之一 Jack Dorsey 宣称，移动支付系统 Square（图 1-9 所示的是通过 Square 网站用 iPhone 进行移动支付的截图）能把智能手机当移动钱包使用。应用中的位置检测功能不仅可以显示交易费用还能显示交易发生的确切位置。

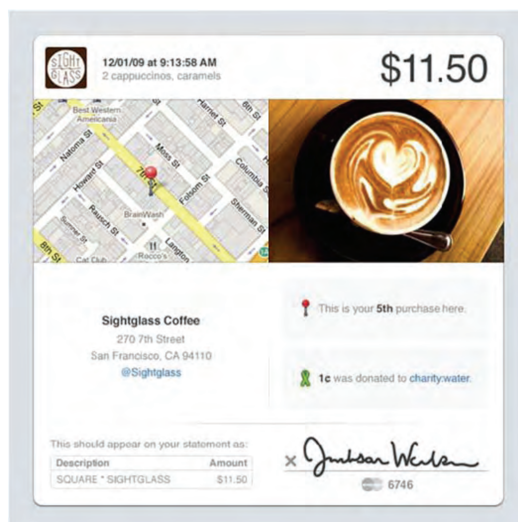


图1-9　Square支持用户使用智能手机及其专有的附加设备（或电子狗）像用传统的商业账户那样用信用卡交易。截图显示的是用户在iPhone手机上进行电子支付

我们将在第 5 章围绕移动应用中的位置全面介绍 LBS 应用的类型，为开发者创新揭示各种可能性。

1.5　小结

基于位置的服务将塑造移动领域的未来，将目前的 Web 1.0 时代推进到全数据全方位交互的 Web 2.0 时代。

对于聪明的开发者和创新型的公司而言，若是能够充分利用用户对 LBS 不断提升的兴趣与期望以及手机科技的重大进步，机会将无可限量。

成功的关键是能够清晰地定义服务的类型、对象以及地点。这就是指：

- 在实现环境感知应用方面的想法要独特，要大胆；
- 迎接移动开发中的各类技术挑战，其中最重要的是应对大多数用户手机屏幕偏小的问题；
- 消除用户对于其位置隐私泄露日益增加的恐惧；
- 选择正确的定位技术或逐步利用优化的混合定位技术。

下一章我们将会了解目前可用的定位技术，既有较新的技术（比如 Wi-Fi 定位），也有较成熟的技术（比如 GPS），以及如何最大化 LBS 的可用性（也就是市场成功的机会）。

第2章 定位技术

本章内容

- ❑ 什么是定位技术
- ❑ 基站三角测量
- ❑ 卫星导航或 GPS
- ❑ 小区识别码（Cell ID）
- ❑ 无线定位系统（WPS）
- ❑ 新兴的定位方法

前一章我们概括介绍了 LBS，你已经知道 LBS 是什么以及为什么移动用户对它有着如此惊人的需求。我们也介绍了定位技术是所有 LBS 应用和服务的关键。本章要详细地了解一下定位技术，以便根据你所开发的应用选择正确的技术。

为了理解选择定位技术的重要性，我们举个简单的例子。假设你开发的应用是利用 GPS 卫星导航来确定用户的位置，并将该位置信息不断显示在城市地图上。用户将该应用程序下载到手机上，激活定位功能，但半个小时后就忘了这件事。过了一段时间，他翻开手机，发现手机没电了。这可不是很好的用户体验，对吗？

仅在几年前，对多数用户来说，移动设备上的 LBS 功能会经常出现这种状况。除了智能手机之外，对于一般的移动设备而言，解决这个问题的方法就是设计一种程序，能将小区识别码作为默认的定位方式（用户可以换成 GPS 方式）。这样就能延长电池的使用时间，以便用户正常使用手机。

这一章，我们会看看使移动设备自动*位置感知*所涉及的全部关键定位技术：基站三角测量、卫星导航定位、小区识别码以及无线定位。

首先了解一下目前移动应用程序中常用的定位技术，然后再讨论新的定位方法[①]。新定位方法包括混合定位技术、P-Cell（参数 Cell）技术以及惯性测量单元（IMU）技术。我们也会初步

① Less widespread or niche positioning technologies include Uplink Time Difference of Arrival (U-TDOA), Angle of Arrival (AOA), and Enhanced Observed Time Difference (E-OTD). You can read more about these at http://en.wikipedia.org/wiki/Mobile_phone_tracking.

了解一下推拉定位（push versus pull positioning）技术的基本原理，因为它会影响用户的位置隐私，第 10 章会进一步讨论隐私问题。

对开发者来说，知道用哪种方法或哪种混合方法非常重要。原因在于他们选择的定位方法会影响移动应用的性能，有时需要在用户定位精度和其他因素（比如位置拟合速度或电池消耗）之间权衡取舍。

智能手机采用混合定位技术克服了很多问题，能够传送精确的位置信息又比较省电。然而，虽然相较于标准手机已进步了很多，但是在智能手机上连接活跃的 GPS，耗电量仍然很大。

虽然智能手机（比如 iPhone）越来越重要，约占据全球手机市场总额的 20%，但是标准手机仍占剩余的 80%。问题在于：智能手机定位能够通过标准 API 获得，很多开发者却不敢冒险跟进。标准手机的开发者会根据手机定位能力优化的程度来开发基于位置的应用。

在开发程序时要权衡定位精度、获取准确位置信息的成本及延长电池使用时效之间的关系。为了让开发者更容易找到这种平衡关系，本章中的每个部分都总结了移动应用中不同定位技术的优劣。

2.1　什么是定位技术

定位技术（在第 1 章里所说的定位组件）是指内置于移动设备中，及时确定该设备所在位置的技术（精确程度不同）。这项技术结合了硬件（比如一个 GPS 芯片）和软件（比如能够从基站信号中读取位置信息的程序）。

所有定位技术都有各自的优缺点。所以 LBS 应用开发者最常用的方法就是在各个移动开发平台上使用本机定位 API。这些 API 一般都能提供经过优化的定位信息（如前面所提到的，经常集成混合定位技术）。

这一部分的目的就是让你了解目前常用的四种主要定位技术：基站三角测量、全球定位系统（GPS）、小区识别码（Cell ID）和无线定位系统（WPS）。我们将会特别介绍不同定位方法的优点和缺点。这样开发者就能做出正确的选择，向移动用户提供最好的 LBS 应用。

在看第 1 章已介绍过的主要定位技术之前，有必要了解一下最早的基站三角测量方法的原理。虽然这是一项老技术，但仍是备用位置服务的基础。三角测量原理也是一些现在常用定位方法的支撑。了解其原理也有助于理解所有其他定位方法的运作机制。

2.1.1　基站三角测量

基站三角测量利用已知的无线电信号传输速度（常由移动电话的超高频频率发出）计算与接收器[①]之间的距离。从几何学的角度来说，通过记录一个物体与其他三个点的距离，就能够计算出这个物体的位置（实际上，这个原理是早期计算地球与月球间距的基础）。

---

① Chris“Silver”Smith，“Cell Phone Triangulation Accuracy Is All Over the Map，”September 22,2008，www.searchengine-land.com.

接收器或天线可以安置在基站上或市区较高的建筑上。至少需要三个最好是四个接收器才能进行较好的位置固定。因为在人口密集的地方基站数量更多，基站信号半径相互重叠，所以位置拟合准确度会很高（能达到 200 米或 700 英尺的精度）。图 2-1 展示了三个基站信号重叠用于计算手机位置。

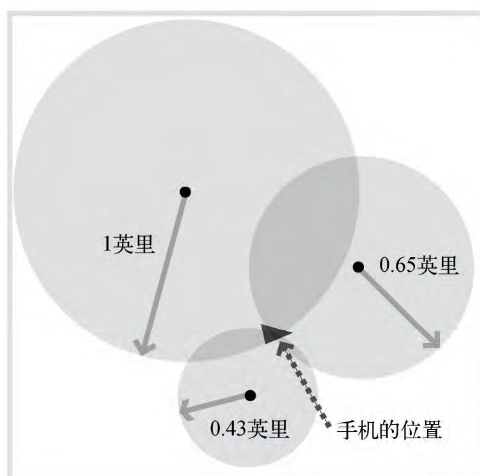


图 2-1 基站三角测量是通过检测三个基站半径各自与手机的距离来进行的。手机的位置就在三个基站半径的重叠处（图片蒙 Chris“Silver”Smith 惠允使用，mng.bz/OQf3）

在基站上安装定向天线会进一步增加位置拟合的准确性，不仅能检测手机信号的距离，还能检测其方位。

在农村安装的传输天线密度很低，仅由一个天线获取基站信号，因此定位精度就会大幅下降（能下降到几公里）。

由于基站三角测量是一项基于网络的定位技术，所以需要与移动运营商签署协议以供移动服务使用。在美国，第 E-911 法规规定，移动运营商有义务协助政府部门在紧急情况下利用基站三角测量确定手机用户的位置。虽然存在一些利用基站三角测量的商业服务（比如 Telmap 之类的移动导航服务），但是在实践中由于费用和运营商的限制，目前多数的 LBS 更乐于采用其他技术，比如 Cell ID、GPS 或者 Wi-Fi 定位技术（或越来越多的混合利用这三种技术）。

2.1.2 卫星导航GPS

GPS 最初是由美国军方在 20 世纪 70 年代研发的。20 世纪 90 年代中期，经私人企业游说后，美国国防部才同意将其作为商用目的。他们也看到了将这项技术民用后的巨大潜能。即使如此，民用版本 SPS（标准定位服务），比留做军用的版本 PPS（精确定位服务）精度要低很多。具有讽刺意味的是，一些企业试图提高标准 GPS 的精度，即弥补被美国国防部蓄意降低的精度。

对于多数的 LBS 应用而言，由于标准 GPS 仍然用普通的定位方法，所以我们先要了解一下

这个方法，然后再去看看在需要更可靠的 GPS 位置拟合时怎样使用 A-GPS（辅助 GPS）。因为电池时效仍是使用 GPS 的主要问题，所以这一节将会提出一种运行 GPS 后仍然保留移动设备电量的方法。

对移动开发者而言，现在 GPS 仍是商用的精度最高的定位技术（最高精度可达 4 米，这个精度足够在一个卧室大小的半径范围内确定设备的位置）。另外，用户可以免费使用 GPS 定位功能，唯一的成本就是已包含在设备价格中的 GPS 芯片。通过了解 GPS 及其变体（像 A-GPS）是如何工作的，开发者就能掌握目前所用的主要定位技术了。

2.1.3　标准GPS

全球定位系统，简称 GPS（仍由美国国防部控制），利用绕地球运行的 31 颗人造卫星组成的星群来进行定位。通过计算从不同卫星到达接收器的时间信号差来确定设备位置。这些卫星按特定轨道高度绕地球旋转，就像空中的参考点（如图 2-2 所示，显示的是卫星 GPS BIIA-28 某天的预定轨道）。

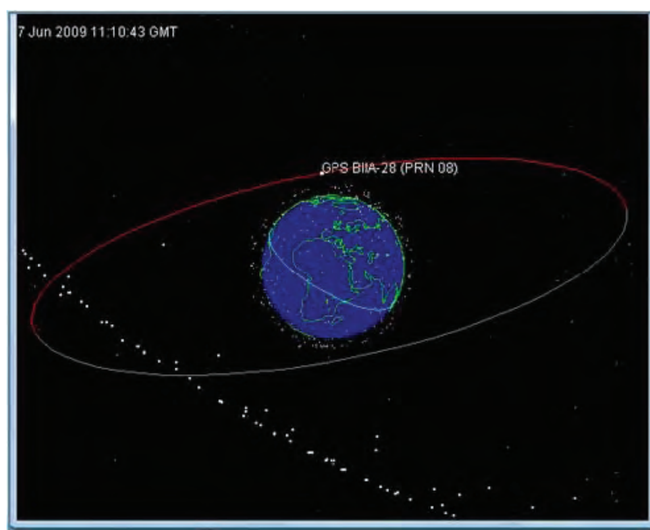


图 2-2　按预定轨道为椭球的绕地球运行的 GPS BIIA-28 卫星（蒙 NASA 惠允使用），于格尼威治时间 2009 年 6 月 7 日早上 11:10

GPS 导航器利用三边测量技术确定用户的位置、速度和高程。GPS 接收器不断接收从 GPS 卫星上发出的无线电信号，并对其分析，计算出与每个跟踪卫星的确切距离（范围）。考虑到卫星在轨道上运行的速率和在地球上检测出该信号从发出到被接收所用的时间，就可以对距离进行数学运算了。

从单个卫星上发出的数据使手机位置限定在地表上的一个较大的范围之内。从第二个卫星上添加的数据使位置缩小到两个球体叠加的区域。从第三个卫星（如图 2-3 所示）添加的数据提供

相对精确的定位信息。从第四个卫星（或更多卫星）上添加的数据提高了定位精度，也能确定接收器的准确高程，或者说高度（就航空器来说）。GPS 接收器一般能同时跟踪四到七颗或者更多的卫星。

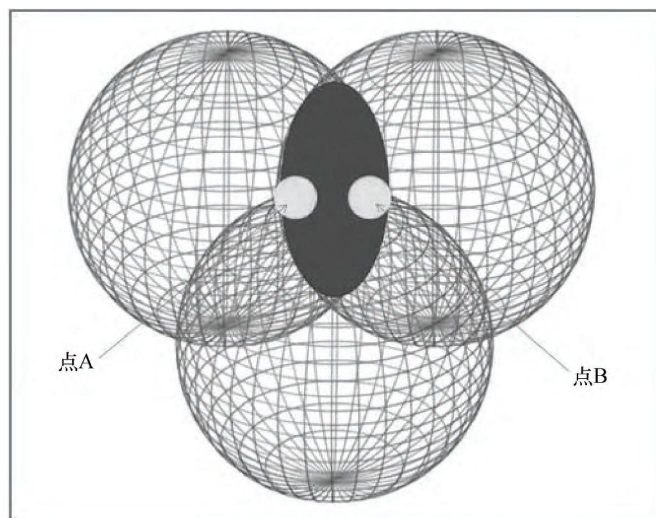


图 2-3　通过三个不同卫星的三边测量，将手机位置缩小到一到两个点上，就是三个球相交的地方，即点 A 和点 B

如果 GPS 导航器接收的卫星数据不足（不能跟踪足够数量的卫星），它会告知用户而不会提供错误的位置信息。虽然 GPS 可能是最精确的定位方法（如果天气晴好，GPS 接收器精度可达 4 至 40 米），但仍有很多弊端：使用该功能时手机电量消耗较大，从冷启动到对可见卫星信号进行初始拟合，需要一段预热时间。在城里使用时会受到“峡谷效应”的影响，接收的卫星信号总是断断续续的。若卫星信号从较高的玻璃建筑物外墙上反射回来，也有多路径偏差，造成位置漂移[1]。

在移动应用中移动设备利用 GPS 进行定位，是相对比较简单的。附录 A 中的代码清单 A-1 的 Java 代码，就是在限制电量消耗的情况下用 GPS 定位的例子。

介绍过了标准 GPS，接着看一看辅助 GPS 是如何提高 GPS 定位拟合精度的。

2.1.4　辅助GPS

由于 A-GPS 部署（根据芯片制造商的配置情况、当地立法以及运营商协议条款）的方法多种多样，与 GPS 相比定位效果也不尽相同，所有至今也没有这项技术公认的定义。

A-GPS 从根本上就是为了解决 GPS 技术不可避免的缺点，换句话说，在户内或密闭的环境中 GPS 不能够定位。而 A-GPS 之所以存在的基本前提就是辅助内置于手机中的 GPS 芯片在不好

① TrimbleGPS Tutorial (www.trimble.com/gps).

的环境（比如卫星信号很微弱或能收到信号的可见卫星只有两颗而不是满足定位要求的三颗的情况）下更快或更准确地确定位置。

就像前面提到的，GPS 芯片在不断扫描天空中相关卫星的轨道运行及数据。扫描结果被称为 TIFF，或者说首次定位时间，也就是 GPS 接收器从开机到定位所需的时间。初始 TIFF 常称作**冷启动**，对于 SiRF Ⅲ系统（最新款的 GPS 系统）来说，仅需 30 秒至几分钟即可在任何地方获取卫星信号[①]。

手机使用 A-GPS，冷启动时间会更快。通常蜂窝网基站都装有 GPS 接收器（或附近有基站），这些接收器不断从卫星上接收星历[②]数据并进行计算。接收的数据被传送到手机上（有要求的情况下），手机像个"骗子"，因为这些位置并不是由它确定的，而是由相关的卫星早就确定好了。处理 GPS 数据的任务要么由第三方服务器要么就通过手机内置芯片（下载星历数据，插入位置固定程序，不需要更进一步的数据网络运算就可以缩短处理时间。）来完成。将手机中断断续续接收到的 GPS 数据（比如由于可视卫星数量较少）与网络辅助服务器数据做比对，就可以获得更精确的位置计算结果了。

鉴于 A-GPS 还是一个相对较新的成果，目前仅安装在小部分用户的手机上。但是大多数的移动制造商正部署将其作为标准配置安装在所有具有 GPS 功能的手机上。

对 LBS 的开发者而言，A-GPS 能用于增强程序支撑力度，因为它能实现快速定位，也能节省电池消耗。问题在于 A-GPS 的安装实施会因为运营商和制造商的不同而变化，需要做扩张分析和试验。还要注意的是 A-GPS 通过移动运营商网络来传输数据（位置），因此移动用户就需要支付费用（而 GPS 的使用是免费的）。

与 A-GPS 实施标准相背离的一个有名的例子就是诺基亚。它在所有新的 GPS 模块上提供 A-GPS 功能，而完全绕过了运营商。诺基亚开发了自己专属的基站定位数据库，在其 Ovi Map 软件（预装于所有具有 GPS 功能的诺基亚手机中）中嵌入基站三角测量程序。

很重要的是长期历书或历书参数常用在手机上。要求 A-GPS 仅隔几天（在更高级的版本中隔几周）就装一次是非常困难的。iPhone 就是这种技术的一个重量级用户，它利用了 Hammerhead 前端/基带和 Global Locate IP。

附录 A 中的代码清单 A-2 中就是手机装置中的 Java 代码，Java 应用在手机使用 A-GPS 功能时就能限制耗电量。相比为标准 GPS 定位设计的代码，仅多了一行代码。

介绍完移动设备的 GPS 定位技术之后，我们再来看另一种获得 GPS 位置信息的方式，这种方式可以在使用外部设备的移动应用中采用。

2.1.5　蓝牙或独立式GPS

前面曾提到移动设备上 GPS 的主要缺点就是电池消耗很大。一个解决方法就是使用一个独立的外部 GPS 设备，通过蓝牙（例子如图 2-4 所示）连接手机。

① Malatesta, "GPS vs A-GPS:A Quick Tutorial," January28,2008,www.wmexperts.com.

② 星历数据中包含任意给定时刻太空中天体的位置信息。



图 2-4　Belkin Bluetooth GPS 设备，能连接移动电话提供额外的 GPS 功能

这种方式的优点在于，即使是没有内置 GPS 的低端移动手机也能无线连接外部设备，从而具有定位功能。GPS 接收机的质量会更高，因此能够确保更快更精确的确定位置信息。另外，其电池寿命要比内置 GPS 的移动设备长很多。

很明显这种方法的主要缺点就是限制了可移动性，需要同时携带两个设备而不是一个。这也是为什么外部蓝牙 GPS 设备更常用在机动车上，配对一部手机使用，成为车内导航系统的另外一种方法。值得注意的是这种外部设备不能连接某些智能手机，比如 iPhone。

GPS 优缺点总结

需要移动用户花费很少的 GPS 已成为一种主流技术，也就是说它仍是 LBS 应用程序中最普遍的定位技术。表 2-1 总结了 LBS 应用中 GPS 定位方法的优点和缺点。

表2-1　用于位置感知移动应用程序的GPS定位技术的主要优点和缺点

优　　点	缺　　点
在最佳条件下，最高定位精度可达4米	导致电量快速耗竭
技术成熟，久经考验	要求空中有一定数量的可视GPS卫星（也就是在户内不能运行）
被手机制造商广泛采用	在某些城市环境中，卫星信号会偏离，导致位置漂移
移动用户无需支付额外费用（只对A-GPS应用收费）	相较其他定位技术，首次启动时间慢

现在我们已经了解了 GPS 定位技术，知道这种定位技术的优缺点，下一步将看看 Cell ID 定位。Cell ID 定位是 GPS 定位的有效补充方法，主要因为其有限的电池使用量。了解 Cell ID，开发者可以将这种技术应用到公共交通工具上。多数情况下，由于这种定位技术耗电量较小，因此就会更实用。

2.2　小区识别码（Cell ID）

在了解 Cell ID 的定义之后，我们会考虑 Cell ID 数据库形式的 Cell ID 信息来源，包括收费的和免费的。并清楚地列举出使用这项技术的主要优点和缺点。

Cell ID 是过去几年里非常流行的定位方法。现在我们要看一看是什么促进了这项技术的发

展，它是如何工作的以及使用这项技术的优缺点。

Cell ID 定位通过使用小区服务基站（向移动设备传输或接收数据的基站），或小区号以及其他已知的位置来寻找移动设备的位置。Cell ID 定位方法很简单，能用在网络或用户设备上，不需要任何升级。

编程提示：GPS 以及轮询频率

由于移动应用中使用 GPS 的主要缺点是电池耗费问题，所以程序员必须认真考虑其位置固定中要求的 GPS 轮询频率问题。通过减少轮询频率，比如说，每 30 分钟一次而不是持续的轮询，就会显著节省电量。缓存用户的最后位置也是这种策略的一种补充方法。这样根据有关地点，位置细节就能在获取到下一个位置前显示出来。

国际电信联盟（ITU）——联合国政府间固定和移动电信管理机构，给每个国家分配了一个移动国家代码（MCC），给各个国家的蜂窝网络运营商分配了一个移动网络代码（MNC）。各个运营商负责为其网络创建位置区码（LAC），并给每个小区分配一个识别数字（Cell ID）。只要移动终端连接到网络就会与其中的某一个基站建立起联系。因此，一个终端的绝对位置就能由 Cell ID、LAC、MNC 以及 MCC 这四个参数表示。

当前的 Cell ID 可以用来识别与移动设备联系的基地收发站（BTS）及其位置。很显然，这种定位方法的精度有赖于基站覆盖的大小，可以相当准确。GSM 网络基站可能分布在任何地方，直径可达 2 公里至 35 公里。利用单一基站确定的位置精度通常在 1 至 2 公里的范围之内。其他与 Cell ID 联合的定位方法精度可在 150 米以内。

在移动设备上使用 Cell ID 定位技术的著名的用户就是 Google。手机上使用的谷歌地图就是利用单一基站传输提供手机位置。这种定位方法经常会导致城市和郊区定位精度不同的现象，如图 2-5 所示。在纽约市区，手机基站之间的半径大小也就是几个街区而已，但是在农村小镇 Slater，基站半径就能扩大到几公里。

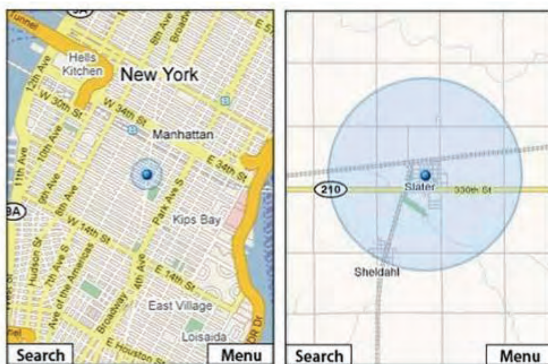


图 2-5　利用谷歌移动地图对比城市和郊区的基站位置检测精度。在纽约市区定位精度在几百米之内，而在像 Slater 这样的乡村小镇，定位精度降低到几公里

Cell ID数据库

要检测 Cell ID 的位置，必须将检测到的运营商小区代码号映射到包含这些代码精确位置的数据库。移动网络运营商并不向外公开自己的小区号信息，也不会提供 Cell ID 数据库的访问接口，原因有很多，其中就有隐私问题，但是更重要的一个原因或许是商业考虑（运营商计划对访问数据收取费用）。

定位技术的编程提示

只要是在移动应用中内置定位功能，都要考虑不同的定位技术导致的程序设计问题。比如，鉴于 GPS 的成熟度和相对广泛的接受度，当然还有移动应用的代码编写，利用 GPS 定位相对简单。Cell ID 和 Wi-Fi 定位技术都有自己额外的发展瓶颈。比如像诺基亚这样的一些原始设备制造商（OEM）并不允许 JME 应用直接从他们的手机中获取 Cell ID 信息。这些问题将会在第 4 章详细讨论。

谷歌地图拥有自己的 Cell ID 数据库，人们普遍认为它的数据库是世界上最完整的。能达到这种精度，主要归功于过去三年中移动设备广泛采用谷歌地图，并在移动用户使用该服务时获取其位置信息以改善原始的 Cell ID 数据库。

除了 iPhone 和 Android（内置了谷歌地图作为地图组件，因此或多或少地强制用户使用其 Cell ID 数据库）之外，移动应用还有很多其他可用的数据库。

很多企业建立了自己的 Cell ID 数据库以供第三方使用。比较有名的是 Navizon，它能提供相对完整的全球 Cell ID 数据库，并且价格合理（在本书写作之时，其企业定位系统解决方案合并 Cell ID 的收费标准大约是每月 100 欧元）。

渐渐地，就 Cell ID 而论，人们对开源解决方案的要求越来越多，并引起了 OpenCell ID 运动。

OPENCELLID

OpenCellID 运动是由一个法国团队领导的开源项目，从 2008 年开始获得显著成效。由于全世界很多人向其提供 Cell ID 的位置信息，据说目前它已经绘制了超过 60 万个基站[①]的位置。

利用很多的公共 API 就可以使用开源项目中的数据了，这些数据有以下功能：

- measure/add
- cell/get
- cell/getMeasures
- cell/getInArea
- measure/uploadCSV
- measure/delete
- measure/list

① As of February 25, 2010. Source: www.opencellid.org.

附录 A 的代码清单 A-3 举例说明了一个 API 密钥反馈用户确切位置信息的样例格式。

有关这些 API 更多的功能细节以及可用 API 的关键问题都在http://www.opencellid.org/api上。由于 OpenCellID 数据库所含的信息遵守创作共用协议，因此是对所有开发者开放的免费资源。值得提醒的是，虽然随着时间的推移，OpenCellID 数据库必然会不断完善，但是现在它最多只能提供拼凑的、并不完整的覆盖信息。有很多国家，像是英国，Open Cell ID 数据库包含了大概 4.5 万个基站（不到该国总数的 50%），但在其他国家，比如西班牙，仅有 6300 个基站的信息，占不到该国基站总数的 25%。

表 2-2 总结了作为 LBS 应用定位方法的 Cell ID 的主要优点和缺点。

表2-2　用于位置感知移动应用中Cell ID定位技术的主要优点及缺点

优　　点	缺　　点
可用在大部分手机上	为确定基站位置须访问Cell ID数据库；访问优质数据库要收费
有赖于标准移动运营网络的连接，能在户内使用	由于定位有赖于单个基站，定位精度会受影响，精度不是几米而是几公里
节能，对电池耗电影响不大	基站发射塔分布稀疏的地方，定位精度低，比如郊区，进一步限制了其实用性

说完 Cell ID，我们要看一下最新的定位技术——无线定位系统。

2.3　无线定位系统

由于无线定位系统（WPS）是定位技术领域中相对较新的技术，所以人们都觉得它很神秘，对于它是如何工作的有些许误解。

这一章将会消除误解，对这项技术给出一个清晰的概述，也会告诉开发者使用这项技术的好处（是的，除了好处还有一些缺点）。

WPS 的主要优点就是能在户内工作，这是传统的 GPS 定位不可能实现的，因此对很多移动应用来说是一项必备的技术。原因在于 GPS 定位需要一定数量的可视卫星，在户内自然不能定位。除了标准 Wi-Fi 之外，出现了一些其他的无线网络标准，比如 WiMAX，但是定位系统的原理是不变的。

Wi-Fi 定位技术需要无线网络传输来保障吗？

人们对 Wi-Fi 常有一种误解，由于 Wi-Fi 路由器经常加密需要口令保护，因此 Wi-Fi 定位技术就行不通。（如果信号被加密还怎么锁定呢？）实际上，Wi-Fi 定位技术不需要连接无线网络，而只是需要获取无线传输的 ID。然后将这个 ID 交叉比对全球数据库，用以确定设备的位置。

Wi-Fi热点检测

Wi-Fi 定位技术是由 Skyhook Wireless 以商用为目的在 2005 年率先研发的。自从作为主要定位技术纳入 iPhone 2G（在内置 GPS 芯片的 3G 可用之前）后变得愈发流行。

Wi-Fi 定位软件利用无线路由器发出的 802.11 无线电信号确定所有可用的 Wi-Fi 设备的精确位置。在某个社区的移动用户一经运行 Wi-Fi 定位功能，软件就会搜寻接入点。通过筛选出的一些信息，对照参考数据库，计算出用户的位置。人口越密集的有 Wi-Fi 信号的地区，软件的定位精度越高。实际上，Wi-Fi 定位原理与早先描述的基站三角测量原理相同，但是 Wi-Fi 定位用于检测由无线路由器传送的信号，而不是检测运营商的无线电传输信号。

Skyhook Wireless 宣称其定位网络覆盖了北美（如图 2-6 所示）和欧洲超过 90%的城市，精度在 20 米至 40 米，由 1 亿部无线路由器为其数据库传送数据。为开发这一数据库，Skyhook 安排人调查了世界上成千上万的城镇，在每一条街道、高速路以及小巷搜寻 Wi-Fi 接入点和基站发射塔，并精确标绘了这些点的地理位置。

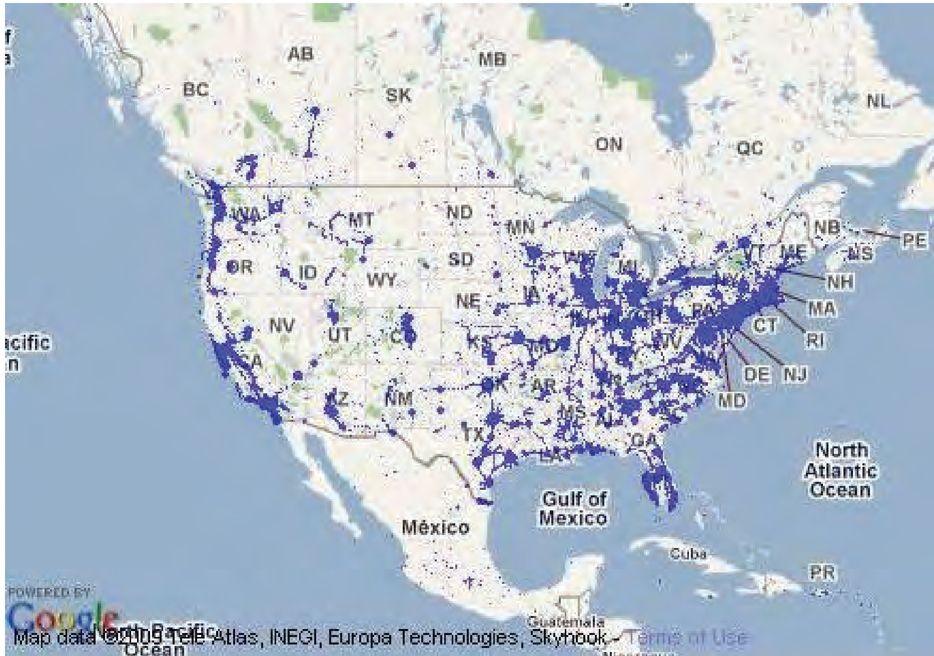


图 2-6 Skyhook Wireless 北美地区数据库的 Wi-Fi 接入点分布图

在这一章中，我们介绍了当今 LBS 应用程序中常用的三种主要定位技术：GPS、Cell ID 以及 WPS。没有任何一种定位技术是完美的，每种技术相对于其他技术都有各自的优点和缺点。因此开发者逐渐开始选择一种混合了三种定位技术的定位方法，叫做混合定位技术。下一节我们就会介绍这种技术，也会考虑定位技术在短期和中期即将面临的问题。

2.4　新定位方法

也许人们会感到奇怪，GPS 最先在商业中使用后的十多年间，定位技术并没有什么进展。不过，20 世纪 90 年代后期以来，它的发展速度正在加快，这主要源于日益增长的 LBS 服务需求和移动运营商门户网站将其作为核心服务来提供。定位技术的关键问题就是提高定位的可靠性，另外，未来的困难则集中在利用移动用户的位置来自动传送基于位置的信息（也包括促销类信息）。

Wi-Fi 以及电池消耗

值得注意的是，就像手机上的 GPS 轮询一样，Wi-Fi 信号检测也会导致电池消耗，虽然程度会比 GPS 稍轻（参见表 2-3）。

表2-3　位置感知移动应用中Wi-Fi定位技术的主要优点和缺点

优　　点	缺　　点
精度高，尤其在Wi-Fi广播密集的城市	要求手机有Wi-Fi功能，这就限制了多数智能手机设备的使用
无需支付额外费用，访问定位需要的Wi-Fi热点数据库是免费的	除了少量高端设备，在移动设备上进行活跃的Wi-Fi扫描有可能使电量耗费很快
位置固定时间短，可以为移动应用提供快速定位	要求服务供应商经常更新Wi-Fi热点数据库。由于数据库变化相对频繁，有可能造成位置固定不准确
由于不需要可视卫星或其他传感器，在 Wi-Fi覆盖区域就能在户内定位	

2.4.1　混合定位技术

混合定位技术是在移动应用中结合几种定位技术（主要是 GPS、 Cell ID 以及 Wi-Fi），以便能够随时传送可靠、准确、经济的位置信息。目前混合定位技术常作为一些智能手机的默认选项，比如 iPhone 和 Android 设备。

知道并利用混合定位解决方案，对于开发者在 JME/Java 平台上部署相关应用很有必要。只依靠单一定位技术进行定位的限制（如表 2-1 和表 2-2 所示）意味着不能像采用混合技术那样给用户一个满意的应用体验。而由于 Wi-Fi 定位方法很难在 JME 平台上实施（原因很多，在第 4 章会详细介绍），混合技术通常结合 GPS 和 Cell ID 定位方法，或 A-GPS 和 Cell ID 相结合的定位方法。

据估计，到 2014 年在所有的定位方案中会有 25%的定位技术是基于混合定位的[①]，然而在一些特殊的移动平台上（个人智能手机平台会是 100%），这个比例很有可能会高很多。

① ABI Research, January 2009; see mng.bz/MxWD.

2.4.2　P-Cell 技术

这项韩国技术试着解决利用常规 Cell ID 技术产生的位置三角问题，也就是运营商网络基站尺寸参数的精度和可靠性。

这项新技术避开常规的 Cell ID 定位方法，而是利用具有规定尺寸的预定义格将某个区域划分成几个部分（如图 2-7 所示），将这种预定义格映射在 P-Cell 数据库中（这里的 P 是“参数化”）。

这种定位技术不断确认是否有位置测量的需求，用户一有需求，它就将从移动终端接收的基本信息与 P-Cell 数据库做比对，确定匹配数据，并将 P-Cell 匹配结果传达给业务用户。该技术的优点就是能将户内位置固定精度提高 70%（至少技术专利人是这样说的）。

该系统工作起来就像标准的基站信号三角测量（更准确地说是三边测量），但不是利用运营商手机发射塔信号半径，而是用有标准尺寸的预设格测量。

在写这本书时，P-Cell 技术（如图 2-7 所示）还仅在韩国本土使用，但是该技术对提高 Cell ID 定位技术的精度有着巨大的潜力。

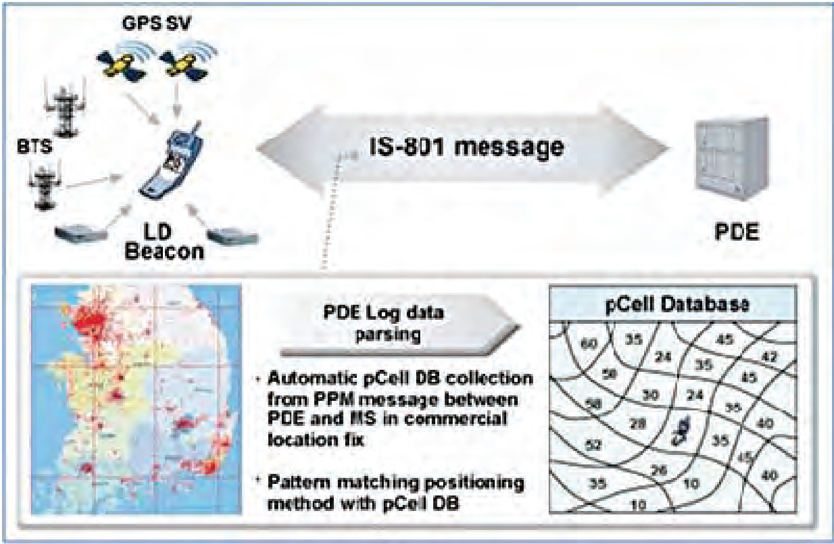


图 2-7　对 P-Cell 技术如何获取移动设备位置的原理描述（感谢韩国 CELIZION 公司提供）

2.4.3　IMU技术

惯性测量单元（IMU）是一种利用加速度计和陀螺仪在航电中测量并报告太空舱速度、方位以及重力的电子设备。IMU 常被用来调遣飞机和宇宙飞船，包括飞船、卫星和登录器。

纳米技术的进展使 IMU 能包装在长度约 1 厘米的部件里，也能安装在智能设备中。图 2-8 展示了一个与安置在 iPhone 和其他智能设备中标准配件相似的加速度计传感器，它在陀螺仪中的使用也越来越多。

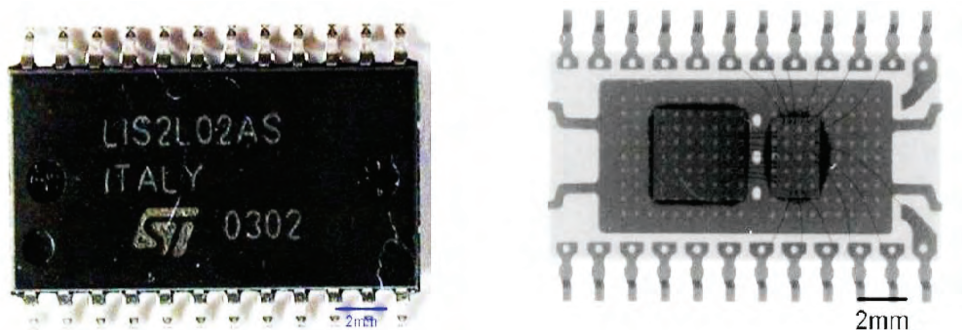


图 2-8　STMicroelectronics 的 LIS2L02AS 双轴加速度计图片。左边的是标准视图。右边是该部件的微组件的 X-射线图像，显示 MEMS(微机电系统)和 ASIC(专用集成电路)用线连接并排安装在一起。有种三轴加速度计安装在 iPhone 和其他智能设备上，能够检测旋转属性，像倾斜、转动和偏航运动（图片来源：mng.bz/kG2q）

IMU 能够根据速度和时间来计算当前的位置，而无须联系外部组件，比如卫星。这种导航方法叫*航位推测法*。这种方法有可能作为越来越受欢迎的定位方法在突然断电的情况下使用，比如在隧道或室内。

值得注意的是加速度计也能用于确定设备是静止或运动。这一点非常实用，它能告知设备是否需要 GPS 重新定位，也能成为位置感知应用的产品特色，因为设备运动可用于触发某些行为（比如，如果运动速度显示用户正在开车，它就能阻断短信的编写，以防产生危险）。

2.4.4　推拉机制

LBS 一项重要的科技和社会障碍就是它能在某个特殊时刻，根据移动用户的位置传送目标信息（比如，促销信息）。

其社会障碍缘于追踪移动用户位置的隐私担忧，在后面会有介绍。技术障碍在于需要使 LBS 应用程序在移动电话上运行，并允许其根据用户位置及预设的触发器（比如地理标记信息）随时启动。这就叫*推送机制*，指自动向移动手机发送通知，不需用户特别请求（另一种信息传输就叫做*拉机制*）。

在标准手机（除了智能手机）上安装这样的程序是不切实际的，电量耗费是主要问题。手机装有 GPS 芯片以及不断搜寻卫星来确定用户位置都会耗费电量。

标准移动电话的另一个解决方法就是依靠移动网络运营商（MNO），利用运营商网络跟踪移动用户的位置。其主要好处就是不必耗用太多手机的电量就可以不断追踪到精确的位置。

实际上，现在的 MNO 对于提供给第三方网络访问接口，进行服务大众市场的（如果真的提供这种接口，很有可能会收费）位置跟踪持谨慎态度。

在智能手机领域，2009 年夏天 OS 3.0 推出后，iPhone 率先采用了推送信息的应用，伴随这

一过程，推送机制的美好前景很可能已经到来。随着 2010 年 OS 4.0 的推出（现在又加入了 Android OS 2.2 及更高版本），推送机制得到了很大的提升。其后台程序可以不断运行，并有望使 LBS 应用以一种简单但有效的方式传递推送信息。

2.5　小结

本章，我们了解了 LBS 应用中使用的主要定位技术，描绘了从最初的定位方法到未来的技术方案的进展。混合定位技术快速成为能够确保 LBS 应用性能优良的新黄金标准。虽然推送机制还没有全面普及，但其被广泛采用，并引导大量刺激消费者的新应用，只是时间早晚的问题。LBS 应用已经召集了全球大量的跟随者，下一章我们会深入了解用户追捧的 LBS 应用。

第 3 章

地　　图

3

本章内容
- ❑ 比较可用技术，选择合适的地图 API
- ❑ 选择合适的地图数据源
- ❑ 在基于浏览器的应用中使用位置

多少个世纪以来，地图一直扮演着重要的角色。地图精度决定财富得失，这句话现在仍然有用。

精确的地图对 LBS 来说是至关重要的。地图供应商以及绘图技术有可能成就或毁掉应用或服务。应用中反映很慢或无应答的地图组件有可能使服务不可用。一个位置细节不足的应用相较于其他详细的地图应用，显得缺乏吸引力。全面了解各种可能的选择，为地图应用挑选合适的组件和技术是非常关键的。

为实现你开发的应用功能，需要选择一个地图 API 和一个地图贴片供应商。地图贴片（tile）是组成地图的小方块图像。虽然多数地图 API 都有自己默认的地图贴片，但是对一些移动平台，可能需要分别从不同来源获取许可的地图贴片。

如果是针对特定地区的服务，确保地图供应商能为该区域提供充足的最新数据非常重要。比如，你在 iPhone 上推出了一款意大利的旅游应用程序，一定要确保能提供最好最详细的意大利地图贴片，以显示意大利所有最新的旅游景点和道路。如果提供过时的地图资料（图 3-1 是一个对比），用户就会卸载你的旅游程序，在应用商店里得到的也将是负面的用户反馈。一次微不足道的失误导致用户流失后，在现今的应用商店中，再想赢回他们就相当困难了。

在过去的十年里，地图业取得了巨大的进步。新出现的 3D 地图有 Google Earth、Bing Maps 以及能对齐各地照片与地图的位置，以供来回浏览的 Street View。将来的进步可能更大，但是正如预期的那样，现在对任何移动 LBS 来说，快速、直观、生动的位置地图或者手机屏上显示的实体是最重要的部分。我们在前面已经介绍了定位技术，这一章将要了解新的地图技术以及市场上主要的地图 API（应用编程接口）。另外还要了解地图在手机屏幕上不同的显示方式，以便能在有限的屏幕上显示真正有用的内容。



图 3-1　不同来源[1]的意大利萨勒诺地图

3.1　地图 API

虽然在线地图 API 有很多种，但是最主要的供应商是 Google Maps、MapQuest、Bing Maps、Yahoo! Maps 以及 CloudMade。

所有这些地图 API 都能可以实现以下功能：创建可视地图、在特定位置使地图居中显示、设置缩放级别、定制标记以及交互控制。所有这些地图 API 也都有各自默认的不同质量和选项的地图贴片。

本节中我们将会讨论优良的地图 API 的一些特性。有些特性多数 API 都具备，有些只有少数 API 才有。但即使是很多 API 都具有的特性，不同 API 的实现也可能大不相同，所以仔细辨别做出正确的选择是非常重要的。决定选择何种地图 API 的重要特性如下。

- 支持 JavaScript。支持 JavaScript 的地图 API 更易与 Web 服务整合。利用 JavaScript API，容易在网站中嵌入互动地图、添加定制数据，能根据用户输入进行动态控制。对象结构、传输速度以及这种 API 是否好用都是开发者需要考虑的重点。
- 支持 Flash/ActionScript。如果想在地图中嵌入动画或 Flex 应用，就要支持 Flash/ActionScript。因为 ActionScript API 与 JavaScript API 是近亲，所以在结果与功能上两者非常相似。
- 3D 地图。像 Google Earth 和 Bing Maps 3D 这样先进的 3D 交互式应用能给用户带来嵌入网页的丰富体验。3D 地图能让用户看三维的建筑物和地形，除了平移和缩放功能之外，还能按某个角度旋转和倾斜。为了达到接近现实的效果，所有的 3D 建筑物都用航空摄影合成。
- 方位。有些地图 API 可以反馈两点之间的驾驶、公共交通和行走方位，并可以用折线（折线就是由一条或多条线段组成的连续的线）显示出来。如果你设计的应用是指引人们从一个地方到另一个地方，这种方位工具就很有效。

① http://compete.com

- 地图视图。默认基础视图是传统的地图视图，能显示道路、高速路和边界。如图 3-2 左边那幅图像。
- 卫星视图。有些地图贴片供应商能提供高精度的卫星影像作为地图贴片，这在一些应用中很重要。而常用的卫星视图，多数是飞机上航拍而来的，并不是卫星影像。如图 3-2 中间那幅图像。
- 地形视图。地形视图重点强调地理特征，显示高程信息，如图 3-2 右边那幅图。
- 混合视图。混合视图是结合传统地图视图和卫星视图的一种视图。在高精度影像图上标记着很多地物特征，如道路和边界。

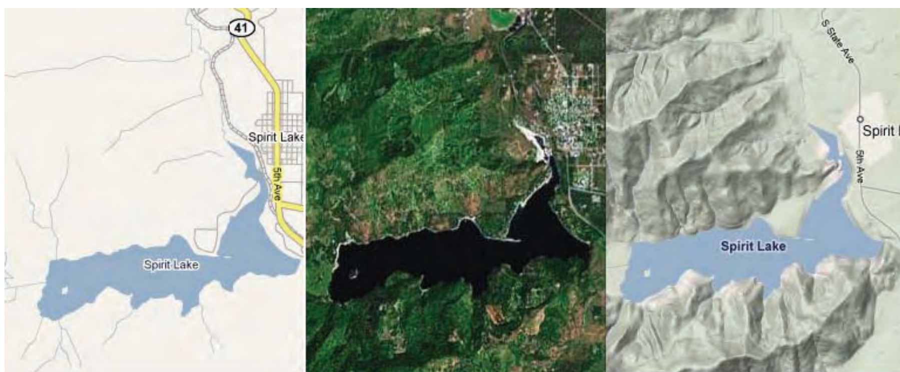


图 3-2 爱荷华州灵湖的地图、卫星和地形视图

- 街道视图。Google Maps 提供的街道视图支持一些主要城市的地平面 360 度视角街道视图。如果向用户显示应用中某位置的实际照片，这种视图特点就可能很重要了。
- 全景视图。只有 Bing Maps 提供，这种视图的特点是能提供美国、加拿大和日本的 100 多个城市以及欧洲 80 多个地方的四个角度的航拍照片。这些正射（常用 40 度倾角拍摄）影像所能提供的细节比直接在建筑上方拍摄的航地图提供的要多很多。在很多全景视图中，能清楚地看到指示牌、广告牌、路上的行人以及其他物体。
- 地理编码。有了地理编码（或地理定位、地理电子标签），就能从其他地理信息，比如街道地址或邮政编码中找到某个实体的地理坐标（经度和纬度）。比如，“1 时代广场，纽约，NY”相当于纬度=40.7566，经度=–73.9862，如图 3-3 所示。

地理编码对任何 LBS 来说都是基础。待会儿我们就能看到支持地理编码的所有地图 API，以及如何在本地服务器或客户端中使用它，以缩短处理时间。使用知名、成熟地理编码 API 的优点就是它们能接收相当多的输入数据（“ny,”“manhattan”，“10001,”“1 park ave.”），并竭尽所能地返回最好的经纬度匹配（或者有时，如果地址不清楚，编码 API 能反馈多种结果，最有可能的经纬度信息优先显示）。如果地址不明确（比如，“1 park ave.”），利用当地的编码数据库就很难或根本不可能获得该位置的确切经纬度。此外，如果地址不对（比如，“1 main street”），那么无论使用哪种服务都不可能标出它的地理位置。所以必须对编码有可能遭遇失败的情况做些准备。

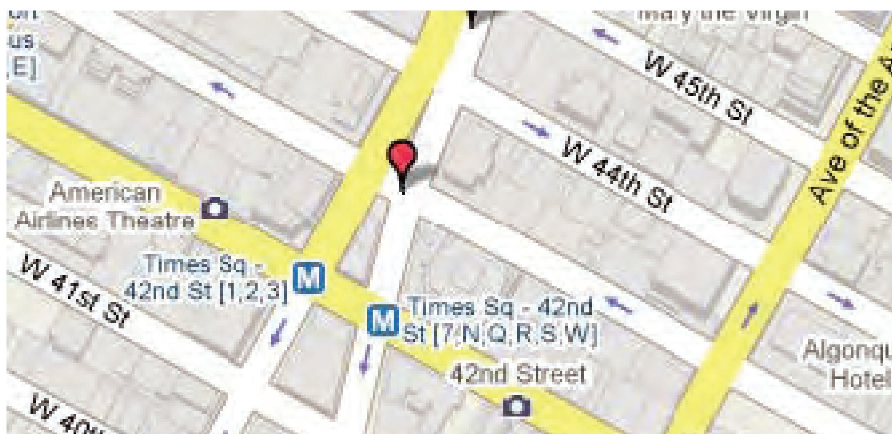


图 3-3 地理编码结果“1 时代广场，纽约，NY”

另外需要注意的是，有时从第三方获得的数据可能只有实体地址但是没有经纬度。例如基于位置的广告就可能只有地址。当你给众多的实体批次编制地理编码时，最好缓存编码结果以减少处理时间。

- 反向地理编码。顾名思义，就是地理编码的相反功能。能够按实体的经纬度坐标找到其位置或邮政编码。反向编码比编码更简单，因为一个经纬度只能匹配一个地址或邮政编码。这种情况下使用本地数据库更合适，可以减少处理时间。
- 适合移动应用。在分辨率较低、显示屏较小的手机上运行地图的话，应选择大字体的注释和少量的说明。必须确保地图供应商能够根据服务运行的地点改变地地图贴片。
- 交通路况。有些地图供应商能为地图提供一些主要城市的实时交通路况。如果线路显示红色，表示严重拥堵路段；黄色表示轻微拥堵；绿色代表一路畅通。
- 企业版。有些地图供应商能提供商业服务软件，其特点就是没有广告，是企业的内部网络。如果想为某个大型企业建设一套独立于公共网络之外的内部网，能控制地图的方方面面，选择企业版的地图 API 是必需的。多数地图 API 供应商要求你的服务也能免费对公众开放。如果你想对用户收费，就需要从地图 API 供应商那儿获得企业许可证。
- 迅捷地图。如果你的应用是互动的，就必须具备迅捷地图（Slippy map）这个特点。迅捷地图是一种能够在 Web 浏览器中使用的地图客户端，用户按住并滑动地图的图片即可动态地平移地图。现代浏览器能根据用户行为动态下载地图贴片，无须重新下载一个页面。这种动态效果使地图浏览更直观。
- 静态地图。如果你的应用不需要互动，那么静态地图影像就会优于迅捷地图。因为使用这种类型的地图能显著减少启动时间和内存使用量，且不需要支持 JavaScript。有些地图 API 提供的静态地图影像能很容易地内嵌到网页上。
- 向量地图。虽然多数地图 API 能提供光栅格式的地图贴片（即位图），基于向量（向量图形是利用如点、线、曲线和多边形这样的几何元素，在电脑图形上利用数学方程式描绘

影像）的地图可能更适用于需要无限缩放的应用，而无须等着下载更多信息。虽然现在这种地图还不能服务于普通大众，但却很有希望。向量地图不需要在每次图像缩放或平移时下载大量的图像文件，它只需要从地图服务器上下载初始数据，然后就能动态生成地图了。这种地图下载速度很快，能非常迅速的显示地图信息。

- 地图样式。有些应用要求地图能改变颜色、字体及其他样式。CloudMade 能提供一种互动编辑器，允许用户根据细节内容自定义地图外观。

3.2　地图 API 比较

在考察了地图 API 的所有重要因素后，我们要更进一步地比较一下各个地图 API 的情况，如表 3-1 所示。这会帮你了解地图 API 的概况。在这一节中，我们也会列出各个地图 API 的一些编码示例，这样你就能通过各个 API 的基本结构及与其他 API 相较它们的优缺点做出正确的选择。

表3-1　地图API比照

特　　性	MapQuest	Google	Bing	Yahoo	CloudMade
JavaScript	是	是	是	是	是
Flash/ActionScript	是	是	是	是	是
3D	—	是	是	—	—
方位	是	是	是	是	是
地图视图	是	是	是	是	是
卫星视图	是	是	是	是	—
地形视图	—	是	—	—	—
混合视图	是	是	是	是	是
街道视图	—	是	—	—	—
全景视图	—	—	是	—	—
地理编码	是	是	是	是	是
反向编码	是	是	是	是	是
适合移动应用	是	是	是	是	是
交通	是	是	是	是	—
企业版	是	是	是	是	是
迅捷地图	是	是	是	是	是
静态地图	是	是	是	是	是
向量地图	—	—	—	—	是
地图类型	—	—	—	—	是

有意思的是，自 2009 年开始 Google Maps 的使用量已经超过了 MapQuest，排名第一，而 Yahoo Maps 仍排在第三位。Bing Maps（最近更改了名称，过去叫 Microsoft Virtual Earth）和 CloudMade

（它刚推出，还不太可能成为用户目的地门户）并没有显示在流量图上，如图 3-4 所示。

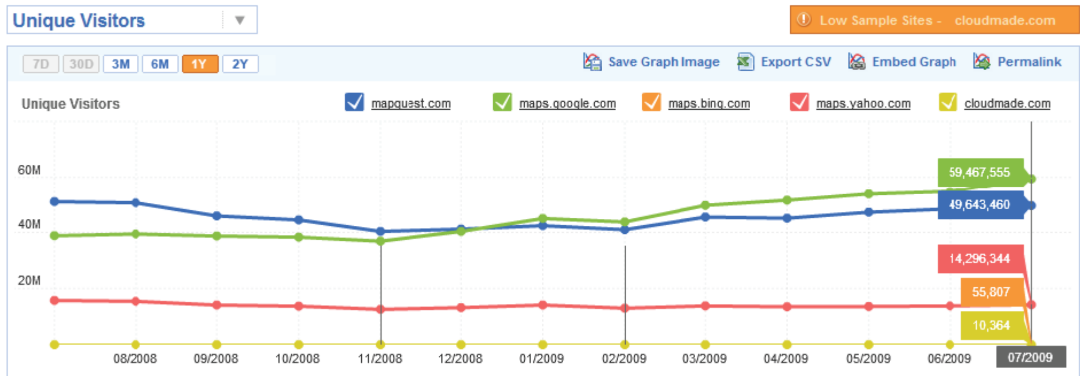


图 3-4 主要地图网站的流量[1]

接下来，我们将分析主要地图 API 的更多细节，提供一些代码段，帮着选择最适合你项目的 API。

3.2.1 MapQuest

MapQuest 是资格最老的一家地图供应商，作为 RR Donnelley 的一个部门成立于 1967 年，制造纸质地图和相关产品。Map Quest 于 1994 年成为一个独立的公司，2000 年被 AOL 收购。MapQuest 网络地图服务于 1996 年推出，并成为当时网上地图的实际标准模式。直到 Yahoo!、Google 和 Microsoft 提供竞争服务后，这种情况才有所改变。MapQuest 在 2007 年 1 月发布了兼容 Ajax 应用的迅捷地图。

MapQuest JavaScript API 是一种现代的面向对象的 API，其命名空间为 MQA，而 MQA.Object 和 MQA.Type 是基础对象。通过创建一个 MQA.TileMap 对象来启动应用程序，然后调用它的方法实现功能。

MapQuest 也有一些引人注目的特征，比如 Icon Declutter，地图视窗中的地图图表相互重叠时，Icon Declutter 能够移开这些图表，保持用户选定的风格。

MapQuest API 适用于 Adobe Flex、 Flash、 ActionScript、 C++、Java、JavaScript 以及.NET，网址为 http://developer.mapquest.com。代码清单 3-1 显示了如何在网页上利用 MapQuest JavaScript API 创建一个简单的互动地图。

代码清单 3-1 MapQuest JavaScript API 代码示例

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
```

① http://compete.com

```
    <script src="http://btilelog.access.mapquest.com/tilelog/
    transaction?transaction=script&key=YOUR_KEY_HERE&itk=true
    &v=5.3.s&ipkg=controls1></script>          <—— 包含 Map Quest 库

</head>
<body>

    <div id="mapWindow" style="width:900px; height:520px;">
    ➥</div>                                  <—
                                                ❶ 声明 div 以容纳地图
    <script language="JavaScript">

        myMap = new MQA.TileMap (                    <—
        ➥document.getElementById('mapWindow'));       ❷ 创建 MQA. TileMap 对象


        MQALatLng(40.0446,-76.4131),"sat");           <—
                                                       ❸ 设置地图中心及贴片类型
    </script>
</body>
</html>
```

这段代码示例首先包含 MapQuest 库，然后创建了一个 DIV❶来容纳地图对象。JavaScript 创建地图对象时，传入 DIV 组件❷。这样就在屏幕上创建了一个地图，由代码设定中心位置和地图贴片的类型❸，由此就能在网页上恰当地显示地图了。

MapQuest 地图上用的 GIS 数据是由 Navteq、 Tele Atlas 以及其他第三方提供的。

3.2.2　Google Maps

Google 在 2004 年收购 Where 2 Technologies 之后，在 2005 年 2 月 8 日推出了在线地图服务。到 2009 年，Google Maps 迅速成长为最受欢迎的在线地图服务。它界面直观、功能丰富，通过 API 被整合在很多 Web 应用之中。

Google Maps API 于 2005 年 7 月发布，提供 Google Earth 整合功能，但是要求安装 Google Earth 插件。2006 年 6 月 14 日，Google Maps 企业版正式推出，作为一种商业服务，它可以在内网运行，并且无广告。2008 年 5 月 15 日，Google Maps API 成功移植到 Flash and ActionScript 3，大大丰富了 Google Maps 的网络应用。

Google Maps API 广泛采用 JavaScript。用户拖动地图，就能从服务器上下载一个小方框，并且小方框会插入到网页上，避免页面的重新下载，因此能为用户提供流畅的应用体验。

Google Maps API 具有最庞大的开发团队，大量的教程以及可以进一步扩展 Google Maps 功能的开放源代码程序库。GMaps Utility Library①就是由 Google Maps API 团队支持的一个辅助程序库。如果要在地图上使用庞大的数据集和标记，就需要这些额外的程序库了。

`GMaps2` 是基本的 Google Maps JavaScript API 对象。这一类对象能在网页上定义一个地图。

① mng.bz/w8rb

在这个类中你可以创建多个实例，每个实例都能在网页上单独定义一个地图。创建完 GMaps2 对象后，可以设置它的属性，调用函数，添加标记，监听用户事件创建互动式应用。

这个 API 适用于 JavaScript、Flash 以及 ActionScript，网址为 http://code.google.com/apis/maps/。代码清单 3-2 显示了如何利用 Google Maps JavaScript API 在网页上创建一个简单的互动地图。

代码清单 3-2 Google Maps JavaScript API 代码实例

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
  xmlns:v="urn:schemas-microsoft-com:vml">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8"/>
<script src="http://maps.google.com/maps?file=api&amp;v=2
    ➥&amp;sensor=false&amp;key=YOURKEYHERE"
    ➥type="text/JavaScript"></script>          ← 包含 Google Maps 库

<script type="text/JavaScript">     ← ❶ 声明脚本
function initialize() {

    if (GBrowserIsCompatible()) {      ← ❷ 检查浏览器是否兼容 Google Maps

        var map = new GMap2(document.getElementById("map_canvas"));

        map.setCenter(new GLatLng(37.4419, -122.1419), 13);

        map.setUIToDefault();          ← ❸ 创建地图对象
    }
}
</script>
</head>

<body onload="initialize()" onunload="GUnload()">   ← ❹ 在页面关闭时调用 GUnload()
    <div id="map_canvas"
   ➥style="width: 500px; height: 300px"></div>      ← 声明 div 容纳地图组件
</body>
</html>
```

这段代码包括了 Google Maps 的程序库和脚本❶。在页面加载后运行脚本以检查浏览器是否和 Google Maps 兼容❷，接下来创建地图对象❸。创建地图对象以后，将其居中显示，利用默认的 UI 小部件设置用户界面选项。

一个重要的细节是，当用户离开了页面后，页面必须调用 GUnload。这对释放 Google Maps 占用的内存至关重要❹。同时，你必须为每个网页设置一个独立的 Google Maps API Key。

Google Maps 使用的 GIS 数据是由 Tele Atlas 提供的。

3.2.3 Bing Maps

2005 年 12 月，Bing Maps（必应地图）以 Windows Live Local 的名义发布，后来改名为 Live Search Maps。2009 年 6 月 3 日，微软正式将 Live Search Maps 更名为 Bing Maps，将 Virtual Earth

平台更名为 Bing Maps for Enterprise。

为保持其在“最新地图详情”上的竞争力，微软不断更新最新的地图数据集；2006 年 11 月 6 日微软在其 Windows 平台上增加了 3D 视图功能。

在 Bing Maps JavaScript API 中 `VEMap` 是一个基本对象。可以通过设置这个对象属性、调用函数来开发应用。

这个 API 适用于 JavaScript（以及企业版的 SOAP），网址为 http://www.microsoft.com/maps/developers。代码清单 3-3 显示了如何利用 Bing Maps JavaScript API 在网页上创建一个简单的互动地图。

代码清单 3-3　Bing Maps JavaScript API 代码实例

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://
     www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">

     <script type="text/JavaScript" src="http://ecn.dev.virtualearth.net/
     mapcontrol/mapcontrol.ashx?v=6.2">
    ➥</script>                                          包含 Bing Maps 库
     <script type="text/JavaScript">
      var map = null;                                   ❶ 页面加载时声明执行
      function GetMap()
      {

         map = new VEMap('myMap');                      ❷ 创建地图对象

         map.LoadMap();                                 ❸ 加载地图
      }
      </script>
   </head>
   <body onload="GetMap();">

      <div id='myMap'
     ➥style="position:relative;
➥width:400px; height:400px;"></div>                     声明 div 对象来容纳地图组件

   </body>
</html>
```

这段代码和其他的地图 API 实例类似。它包含了 Bing Maps 的程序库。当页面加载后调用主脚本❶、创建地图对象❷，接着加载地图❸，这样就可以在网页中显示地图了。

提示　仅通过改变前面的一行代码，就能使用 3D 地图模式，并将地图居中显示在特定的坐标系中。用 `map.LoadMap(new VELatLong(47.22, -122.44), 12, 'r', false,VEMapMode.Mode3D, true)`替换 `map.LoadMap()`。

Bing Maps 使用的 GIS 数据是由 NAVTEQ 提供的。

3.2.4　Yahoo! Maps

1996 年 4 月 24 日 Yahoo（雅虎）在它的网络向导中增加了 Yahoo! Maps 功能，该功能是与 Proximus 公司共同开发的。用户从这些静态地图可以看到未来趋势的一点儿端倪。

Yahoo 不断在地图业务上追加投资，于 2005 年 11 月 2 日推出了互动 Yahoo! Maps。即使如此，Google Maps 和 MapQuest 还是两个最大的在线地图供应商，而 Yahoo!排在其他同类公司之前，位居第三。

在 Yahoo! Maps JavaScript API 中，`YMap` 是一个基本对象。首先要通过创建这个对象的一个实例来启动应用，然后设定对象属性、调用函数来扩展功能。Yahoo 也提供像 Local、Weather 之类的 API，这些 API 能与 Yahoo 的地图 API 相互配合得很好。

这种 API 适用于 JavaScript、Flash 以及 ActionScript，网址为 http://developer.yahoo.com/maps。代码清单 3-4 显示了如何通过 Yahoo! Maps JavaScript API 在网页上创建一个简单的互动地图。

代码清单 3-4　Yahoo! Maps JavaScript API 代码实例

```
<html>
<head>

<script type="text/JavaScript"
src="http://api.maps.yahoo.com/ajaxymap?v=3.8&appid=YMAPSKEYHERE">
    ➥</script>                                   ◁—— 包含 Yahoo!Maps 库

<style type="text/css">                            ◁┐
    #map{ height: 75%; width: 100%;}                 ❶ 设置地图组件的样式属性
</style>
</head>
<body>
                                           ┌声明容纳地图组件的 div 元素
<div id="map"></div>                      ◁┘
<script type="text/JavaScript">                                  ❷ 创建地图对象

    var map = new YMap(document.getElementById('map'));   ◁┘

    map.addTypeControl();                        ◁—— ❸ 添加地图类型控件

    map.setMapType(YAHOO_MAP_REG);             ◁—— ❹ 设置地图类型

    map.drawZoomAndCenter("San Francisco", 3);             ◁┐
</script>
</body>                                                      ❺ 以地理编码位置为中心显示地图
</html>
```

这段代码首先调用 Yahoo! Maps 库，设置地图组件的样式❶，在加载页面后调用主脚本创建地图对象❷，然后添加地图类型控件❸，将地图初始类型设置为常规类型❹，将缩放级别设置为 3，居中显示 San Francisco 地图❺。

Yahoo！Maps 使用的 GIS 数据由 NAVTEQ 提供。

3.2.5　CloudMade

CloudMade 是在线地图领域的新生事物。该公司于 2009 年 2 月推出产品，提供 API 和完全定制的动态实时地图。

CloudMade 是由 OpenStreetMap 的创始人 Steve Coast 创立的，只使用 OpenStreetMap 的 Creative Commons–licensed（创意共享许可的）数据。

作为全功能的迅捷地图，CloudMade 支持 OpenLayers。OpenLayers 是一种能够在线缩放/平移，动态显示地图的 JavaScript 库。这种全功能的 JavaScript API 支持图层选择、透明显示、添加标记、向量对象及其他功能。

CloudMade 还有一个叫做 Web Maps Lite 的轻量级替代产品，适用于那些需要简单，快速加载界面的开发者。

CloudMade 的样式编辑器（见图 3-5）能让用户在地图上编辑颜色和可视图层，还能定制 Google Maps、 Yahoo!Maps 及其他大规模竞争对手不提供的功能。

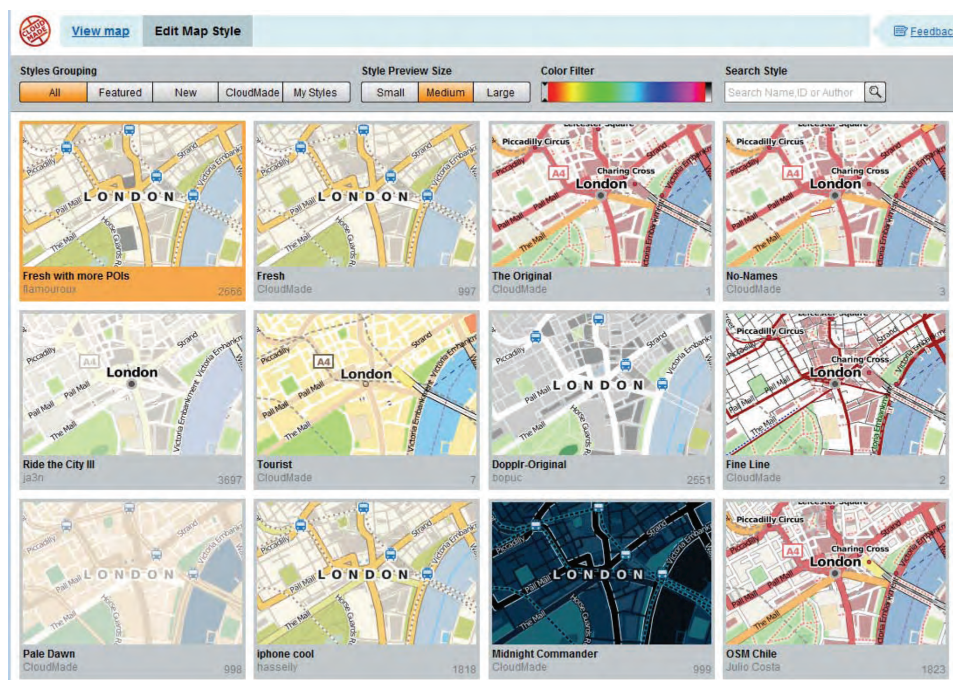


图 3-5　CloudMade 的样式编辑器

CloudMade API 适用于 JavaScript、 Flash、 Python、 Ruby、Java、 Perl、C++、iPhone、Java ME、Windows Mobile、Symbian、Android、BlackBerry 以及 Windows 平台。网址为：http://developers.cloudmade.com。由 Nutiteq 公司提供的 CloudMade 手机 SDK 涵盖了 Android、Java ME 和 RIM 以及 Windows Mobile 和 Symbian 上的 CartoType，但不包含 iPhone，CloudMade 针对

iPhone 有自己的手机 SDK。

代码清单 3-5 显示了如何利用 CloudMade Java Script API 在网页上创建一个简单的互动地图。

代码清单 3-5　CloudMade Java Script API 代码实例

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/
     TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>

  <script type="text/JavaScript" src="http://tile.cloudmade.com/wml/latest/
     web-maps-lite.js"></script>          <- 包含 Cloud Made 库
</head>
<body>
                                          声明容纳地图组件 div 对象
  <div id="cm-example" style="width: 500px; height: 500px"></div> <-

  <script type="text/JavaScript"> <- ❶ 声明加载页面时运行脚本
                                          ❷ 创建主 Clound Made 对象
    var CloudMade =
    ➥new CM.Tiles.CloudMade.Web({key: 'YOURKEYHERE'}); <-

    var map = new CM.Map('cm-example', CloudMade); <- ❸ 创建地图对象

    map.setCenter(new CM.LatLng(51.514, -0.137), 15);    <-

  </script>                                  ❹ 设置地图的中心位置及
</body>                                         缩放级别
</html>
```

这段代码实例首先调用 CloudMade 库，然后调用主脚本❶，加载页面时就会调用主脚本，创建主 `CloudMad` 对象❷，进而用于创建地图对象❸。创建完主地图对象之后，代码会以特定的经纬度居中显示地图，设置缩放级别为 15❹。这样就能在页面上显示地图了。

3.2.6　Mapstraction

可供选择的不同地图 API 供应商有很多，每个 API 在功能、设计和服务条款等方面都略有不同。Mapstraction 允许你轻松选择不同的地图 API，只要编写一次代码，就能根据需要在不同的 API 中进行切换。

Mapstraction 是一个开源库（BSD 许可）。它可以为不同的 JavaScript 地图 API 提供通用 API，以实现地图 API 的轻松切换。开发者一旦编写好代码，就可以根据项目需求、条款和条件以及新功能轻松切换不同的 API。

用户可以根据个人的意愿和当地地图质量切换地图。Mapstraction 有不同的工具，让用户很容易就能在他们的网站上整合地图，设置不同的控件、样式和供应商。

Mapstraction 支持：

- 11 个主要地图供应商；
- 在不同的供应商中动态转换；

- ❑ 点、线和多边形；
- ❑ 通过时间、种类或其他特性进行标记筛选；
- ❑ 影像叠加和基础贴片；
- ❑ GeoRSS 和 KML 信息源导入；
- ❑ 地理编码；
- ❑ 行驶方向。

Mapstraction API 相关内容见http://mapstraction.com，代码清单 3-6 显示了如何利用 Mapstraction API 在网页上创建一个简单的互动地图。

代码清单 3-6　可以在多个 API 中转换的 Mapstraction 代码实例

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/
     TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
    ➥xmlns:v="urn:schemas-microsoft-com:vml">
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8"/>


  <script src="http://maps.google.com/maps?file=api&v=2&key=GMAPSKEYHERE"></
     script>
  <script type="text/JavaScript" src="http://api.maps.yahoo.com/
     ajaxymap?v=3.0&appid=YAHOOKEYHERE"></script>
  <script src="http://dev.virtualearth.net/mapcontrol/mapcontrol.ashx?v=6"></
     script>
  <script src="http://openlayers.org/api/OpenLayers.js">
  </script>                ❶ 包含所有使用的地图 API

  <script
     type="text/JavaScript" charset="utf-8" src="http://mapstraction.com/
     mapstraction-js/mapstraction.js">
    </script>              包含 Mapstraction 库

  <script type="text/JavaScript">        ❷ 声明主脚本
    var mapstraction;
    function initialize() {

     mapstraction =
    new Mapstraction('map_canvas','yahoo');      ❸ 创建地图对象


     mapstraction.setCenterAndZoom(
        new LatLonPoint(37.75,-122.44), 8);      ❹ 设置地图的中心位置
    }
  </script>
  <style type="text/css" media="screen">
    .mapstraction { height: 300px; width: 100%; z-index: 1; }
  </style>
</head>
```

```
<body onload="initialize()"
  style="font-family: Arial;border: 0 none;">      <── 页面加载时初始化调用


  <div id="map_canvas" style="width: 100%; height: 400px">
   </div>                          <──
</body>                               为地图组件声明 div 对象
</html>
```

这段 Mapstraction 代码首先加载所需的库❶。如果不想在地图 API 中动态转换就没有必要在 HTML 代码中包含所有的 JavaScript 库，只要包含要用的那种地图 API 就可以了，这样可以减少启动时间和内存使用量。

接着代码运行主脚本❷，创建地图对象，传入你选择的供应商和 API ❸。在这个实例中，我们使用了 Yahoo! API。Mapstraction 将所有的地图 API 调用传给 Yahoo! JavaScript API，创建一个 Yahoo!地图对象。然后代码实例按特定的经纬度居中显示地图，设置缩放等级为 15❹，这样一个 Yahoo!地图就在页面上显示出来了。这时候，如果用户要求通过另一个供应商来浏览地图，地图 API 就可以根据用户的反馈转换成另一个供应商。

3.2.7　关于许可

在打算利用哪种地图 API 之前，分析它们的使用条款是非常重要的。否则，有可能会浪费大量的时间和精力。

例如，Yahoo! Maps API 使用条款[1]中声明你不能：

使用由 GPS 或其他位置感知设备获取的时间在 6 个小时之内的位置信息；

利用 GPS 或其他位置感知设备获取的，没有直接由最终用户上载到你的应用或服务上的位置信息。

Google Maps API 使用条款[2]中声明你不能：

在浏览器外部使用静态地图 API。

以及你的服务不能：

要求付费订阅或不付费就限制访问。

另外，关于广告：

目前的服务不包括在地图影像上发布广告。然而，Google 有权通过你的服务在地图影像上发布广告，但会提前 90 天告知。

如果你的服务要求付费订阅，在没有获得 Google 明确许可的情况下，是不能使用 Google Maps API 的。如果你的应用需要显示 6 个小时之内的位置信息，那么不能使用 Yahoo! Maps API。移动应用也通常要获得明确许可，因此仔细阅读所有的使用条款，与供应商联系，以确保今后不会遇到麻烦。否则，你最终得到的很可能是一个贴满 Google 广告的 Google Maps 应用。

现在我们来了解一下地理数据公司。

---

① mng.bz/qp4b.

② http://code.google.com/apis/maps/terms.html.

3.3 地图供应商

谈到地图数据源不得不提一下两大商业竞争对手：NAVTEQ 和 Tele Atlas。除了这两个之外，还有 OpenStreetMap 形式的开源产品。

你可以购买地图贴片，也可以利用某个地图贴片服务。比如，可以从微软购买一个服务，通过 NAVTEQ 为你的应用提供地图贴片。也可以从 CloudMade（OpenStreetMap 数据）购买自定义地图贴片，并基于 OGC（Open Geospatial Consortium，开放地理空间联盟）制定的 WMS（Web Map Service，Web 地图服务）标准来用自己的服务器提供这些贴片。

提示 **WMS** OpenGIS WMS 接口标准提供简单的 HTTP 接口，可以用来从一个或多个分散的地理数据库中请求经过地理注册的地图图片。WMS 请求指定地理图层和需要处理的兴趣区。对这个请求的响应是一个或多个经过地理注册的地图图片（如 JPEG、PNG 等格式），可以在浏览器应用中显示出来。该接口也支持指定是否返回透明图像的功能，以便合并来自多个服务器的图层；否则，就无法合并。

有一种使用自定义地图贴片的办法就是使用 GeoServer。GeoServer 是用 Java 编写的开源服务器，用户可以用它来使用或编辑地理空间数据。为实现互操作性，GeoServer 发布使用开放标准的任何开放式空间数据。GeoServer 是 OGC Web 的参考实现，它实现了 Web Map Service 和 Web Coverage Service 规范。

在你的应用中，可以根据地理位置动态切换供应商，为指定区域提供最好的贴片。

提示 **平移和缩放** 在一个速度慢、功率小的移动设备上实现地图的平移和缩放，是一个很常见的技术难题。想在屏幕上显示地图贴片,就应该从供应商那儿获取适合缩放级别的贴片。用户在移动地图时,应该请求现有图片附近的图片。最好按照先进先出的顺序，在内存中缓存那些长时间不用的图片贴片。时间最久的贴片首先应该删除。而在用户放大或缩小地图时，必须替换所有的贴片，在设备内存不足的情况下，可能就得清除所有缓存。对于资源短缺的移动平台而言，管理图片贴片缓存是一个很大的问题。为此，需要不断尝试，才能找到最优方案。

现在，我们将分析一下三个经常看到的主要地图数据源。

3.3.1 NAVTEQ

NAVTEQ[1]的基础数字地图和地图内容，为全球导航和 LBS 产品的发展提供支持。该公司成立于 1985 年，总部设在芝加哥，2007 年被 Nokia 收购，但一直独立经营。

① http://navteq.com

NAVTEQ 的所有数据都是来自第一手观测，而不是政府发布的数据。它提供的数据应用十分广泛，包括 BMW（宝马）、 Chrysler（克莱斯勒）、 Mercedes-Benz（奔驰）、Mini（迷你）和其他汽车制造商使用的汽车自动导航系统（占市场份额的 85%）。

由 Garmin、Magellan 和 Lowrance 生产的便携式 GPS 以及 Yahoo! Maps、Bing Maps 和 MapQuest 等互联网应用都采用 NAVTEQ 的地图。微软的 Flight Simulator X 利用 NAVTEQ 数据自动生成地形图。

NAVTEQ 的电子地图数据库，如图 3-6 所示，目前覆盖了六大洲的 77 个国家。

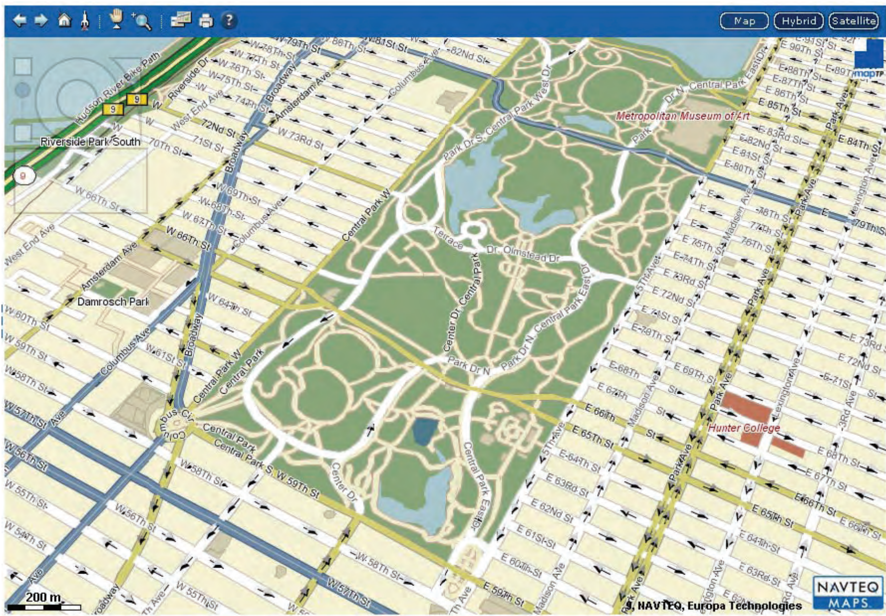


图 3-6　NAVTEQ 自己的 Java 程序显示的纽约中央公园，倾斜角 45 度来自http://navteq.com

NAVTEQ 的主要竞争对手是 Tele Atlas。

3.3.2 Tele Atlas

Tele Atlas[①]是一家荷兰公司，创立于 1984 年。该公司为地图、导航以及包括个人和汽车导航系统的 LBS 提供数字地图和 GIS 数据，提供的数据广泛用于移动和在线地图应用。2008 年 7 月 30 日，该公司成为自动导航系统制造商 TomTom 公司的全资子公司。

Tele Atlas 与 Google Maps 签订了向 Google Maps 提供地图数据的协议，截止年限为 2013 年。按协议，Tele Atlas 有权更正和更新由 Google Maps 社区提供的数据。

Tele Atlas 收集数据要依仗很多不同的资源，包括装有每四秒就拍摄一次公路及公路设施高科

① http://teleatlas.com

技照相机的货车；城市、政府及当地资源；邮政资源。这些资源预示新的发展，相应地，公司按优先级安排信息收集工作。Tele Atlas 声称他们有智能驾驶技术，因此成本控制得很好。

Tele Atlas 提供的地图（如图 3-7 所示），包括了全球 90 个国家的详细街道以及互相连接的公路网络。另外，Tele Atlas Connect 向世界各地有数码地图覆盖的 200 多个国家和地区提供基础地图和路线选择功能，如图 3-8 所示。

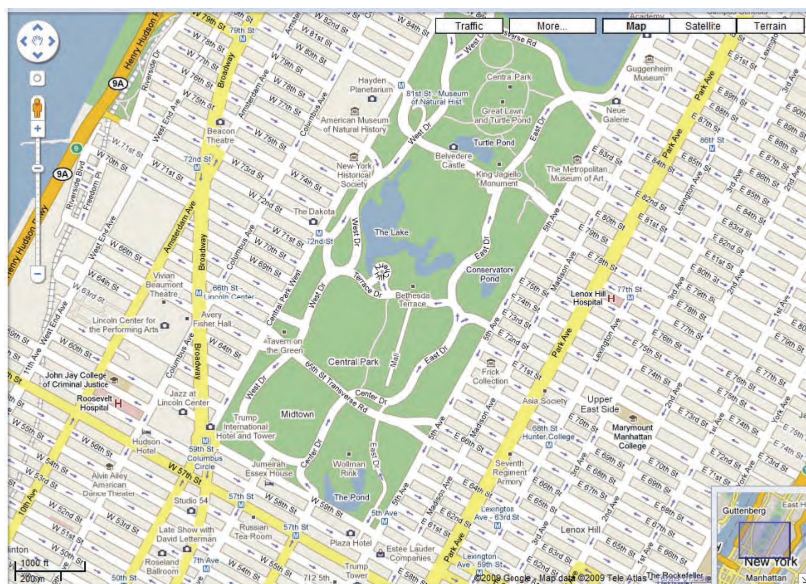


图 3-7　用 Google Maps 浏览纽约中央公园，Tele Atlas 地图数据



图 3-8　Tele Atlas 全球覆盖图（来自 Tele Atlas 网站）

提示　**TELE ATLAS 与 NAVTEQ 地图数据**　Tele Atlas 与 NAVTEQ 的地图数据精度是两家公司及其用户争论的焦点。虽然有些用户认为在美国 NAVTEQ 地图的精度会更高些，但是研究表明两个数据供应商之间并没有很大的差别。

通常，不论 Tele Atlas 还是 NAVTEQ，地图数据质量都会因位置的不同而相差甚远，这主要依赖于他们的本地数据合作伙伴。最好在获得公司数据许可前，根据特定的地点仔细比较不同商家的数据。

3.3.3　OpenStreetMap

OpenStreetMap[1]是一个在线地图协作项目，目标是创造一个能自由编辑的世界地图。其地图数据就像 Wikipedia，任何人都可以创建和编辑。

如图 3-9 所示的地图，是由用户根据手持 GPS 装置、航空摄影照片、其他免费资源或地方性知识绘制而成。渲染图像和矢量数据集都可以在 Creative Commons Attribution-ShareAlike 2.0（共享创意姓名标示–相同方式分享 2.0）授权下下载。

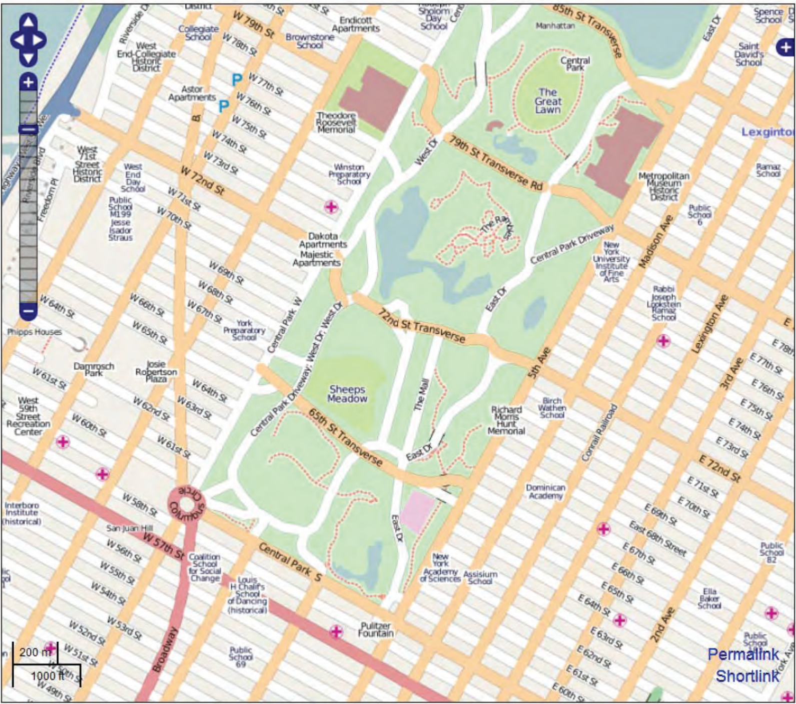


图 3-9　OpenStreetMap 的纽约中央公园数据，网址http://openstreetmap.org

① http://teleatlas.com

OpenStreetMap 的诞生是为了能够创建免费的开源地图，任何人都可以在应用中使用该地图数据，而不用向地图数据供应者支付费用。来自商业地图机构的数据包含版权“复活节彩蛋”。这些复活节彩蛋以各种不同的形式存在，或是找不到的街道，或是根本不存在的教堂和学校。使用许可数据的地图就没有这些问题了。

由于 OpenStreetMap 的内容全部是由公众以 Wikipedia 风格编辑的，因此不保证任何程度的精度，不过它能提供一个起始点，在此基础上不断改进和完善数据的准确性和完整性。

我们已经分析了所有可用的地图 API 和地图贴片来源，现在看一看另一个对 LBS 应用至关重要的要素：找到用户所在位置，为他们提供最恰当的信息。

3.4　基于浏览器的位置

LBS 不再局限于手机或 GPS 设备，运行在浏览器上的 Web 服务现在也能通过集中式数据库（如 Fire Eagle）或通过 Wi-Fi 标志地理位置和 IP 地理编码（如 Skyhook Wireless 的 Loki，以及使用 Google 后端地理位置的 Firefox 3.5）来获取用户位置。

为了给你提供更多的相关信息或节省搜寻时间，利用位置感知浏览网站时会检索你的位置。比如你正在寻找附近的比萨店，网站就可以询问是否能分享你的位置，这样简单搜寻一下“比萨”，就能获得所要的结果。不需要更多的信息或额外的输入。

这种方法已被 World Wide Web Consortium（万维网联盟）标准化为 Geolocation API[①]。这个规范定义了 API 能够让脚本访问用户设备（这里就是指浏览器）的地理位置信息。

在下一节中，我们将会了解一下有关位置的 Web 服务。

3.4.1　Fire Eagle

Fire Eagle[②]是 Yahoo!的一种服务，能够储存用户位置。经用户许可，Fire Eagle 可通过 Fire Eagle API 提供不同的服务。用户可以批准能否写入或读取自己的位置数据，可以利用一项服务在 Fire Eagle 上设置位置，然后在另一个（可能无法直接利用 GPS 的）服务中使用这一位置信息。

比如用户可以在 Fire Eagle 上设置从手机 GPS 获取的当前位置。随后就能通过便携式电脑在一个音乐会网站上使用这个位置。正常情况下无法获取用户位置的音乐会网站，就能利用用户的当前位置显示其附近的所有音乐会，从而提供更好的服务。

Fire Eagle 依靠 OAuth，很多函数库都可以用来访问该服务，对应的语言有 PHP、JavaScript、ActionScript、C#、Java (Android 及 J2ME)、Perl、Python 以及 Ruby。

图 3-10 显示了 Fire Eagle 如何工作。

---

① http://www.w3.org/TR/geolocation-API/

② http://fireeagle.yahoo.net/developer/documentation

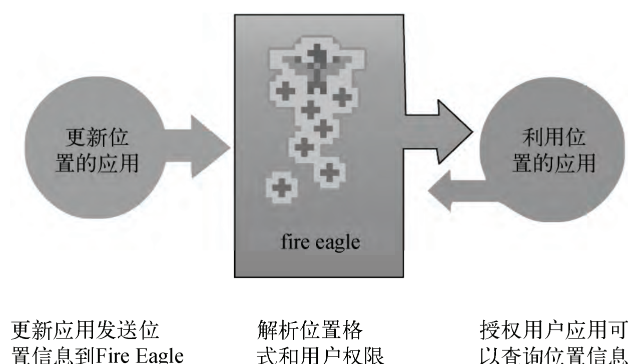


图 3-10　Fire Eagle 概况（来自 Fire Eagle 网站）

3.4.2　Loki

Skyhook Wireless 发布的 Loki[①]是一个免费的虚拟 GPS 工具栏，能自动结合用户位置和 Google Maps、Fandango、Weather.com 以及更多 Web 内容。代码清单 3-7 展示如何使用 Loki 获取用户的位置，并用 Google Maps API 显示。

代码清单 3-7　使用 Loki 在地图上显示用户位置

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:v="urn:schemas-microsoft-com:vml">
<head>

<meta http-equiv="content-type" content="text/html; charset=utf-8"/>

<script src="http://maps.google.com/maps?file=api&amp;v=2&amp;sensor=false
   &amp;key=YOURKEYHERE" type="text/JavaScript"></script>        包含 Google Maps 库
<script type="text/javascript" src="http://loki.com/plugin/
     ➥files/loki.js"></script>        包含 Loki 库

<script type="text/javascript">
//<![CDATA[
                                            ❶ 主脚本
var map;
function init() {                                          ❷ 创建主 GMap 对象

  map = new GMap2(document.getElementById("map"));

  map.setCenter(new GLatLng(42.36463232550283,
                         -71.05836868286133), 13);
                                                          ❸ 在指定坐标居中地图
```

① http://www.loki.com/

```
  requestLocation();
}
```

❹ 发出 Loki 请求

```
function requestLocation() {

  var loki = new LokiAPI();
```

向 Loki 请求用户位置

```
  loki.onSuccess = function(location) {
```

请求成功时调用

```
    var point = new GLatLng(parseFloat(location.latitude),
                            parseFloat(location.longitude));

    map.setCenter(point, 13);
```

❺ 重新居中地图

```
    var marker = new GMarker(point);

    map.addOverlay(marker);
```

❻ 创建新标记

```
    marker.openInfoWindowHtml(location.latitude+','
                             +location.longitude+'');
  };
```

❼ 在标记处打开弹出窗口

请求失败时调用

```
  loki.onFailure = function(error, msg){
    alert('An error has been encountered ('+error+'). '+msg);
  };


  loki.setKey("YOURKEY");
```

把 KEY 加载到 Loki 对象中

```
  loki.requestLocation(true,loki.NO_STREET_ADDRESS_LOOKUP);
}
```

❽ 向 Loki 发送请求

```
//]]>
</script>
</head>

<body onload="init()" onunload="GUnload()">


    <div id="map_canvas" style="width: 500px; height: 300px"></div>
</body>
</html>
```

声明 **div** 地图

这段代码包含了 Google Maps 和 Loki 程序库。主脚本❶建立了 Google Maps 实例❷并在其初始位置使地图居中❸。然后代码调用函数，请求 Loki 查找用户位置❹。如果成功找到该位置，代码就会将找到的位置❺在地图上居中显示，创建一个标记❻，在该位置上打开一个弹出窗口

❼。实际的 Loki 请求通过调用 loki.requestLocation 发出❽。

3.4.3 Geode

Geode[①]是 Firefox 的插件，会查找用户计算机的位置并提供个性化和本地代的内容。Geode 遵从 W3C 的 Geolocation API 规范。

代码清单 3-8 展示了一个网页如何利用 Geode 访问用户位置，提供更多相关信息。

代码清单 3-8 利用 Geode 访问用户位置

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
 ➥xmlns:v="urn:schemas-microsoft-com:vml">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8"/>

<script src="http://maps.google.com/maps?file=api&amp;v=2&amp;sensor=false
➥&amp;key=YOURKEYHERE" type="text/JavaScript"></script>
<script type="text/javascript">
//<![CDATA[

var map;

function init() {

  map = new GMap2(document.getElementById("map"));

  map.setCenter(new GLatLng(42.36463232550283,
                           -71.05836868286133), 13);

  navigator.geolocation.getCurrentPosition( function(pos){
        var point = new GLatLng(parseFloat(pos.latitude),
                           parseFloat(pos.longitude));
        map.setCenter(point, 13);

        var marker = new GMarker(point);

            map.addOverlay(marker);

        marker.openInfoWindowHtml(pos.latitude+','
                           ➥+pos.longitude+'');
     }
  )
}

//]]>
</script>
</head>
```

包含 Google Maps 库

❶ 页面加载后调用

❷ 创建 **GMap** 实例

❸ 居中显示地图

❹ 访问 **navigator** 对象的 **geolocation** 属性

❺ 创建新 **point** 对象

❻ 居中显示地图

❼ 创建标记对象，添加到地图上

❽ 打开标记信息窗口

① mng.bz/37pW

```
<body onload="init()" onunload="GUnload()">

    <div id="map_canvas"
        ➥style="width: 500px; height: 300px"></div>        <— 声明 div 容纳地图
</body>
</html>
```

Geode 实例代码加载 Google Maps 程序库，以主脚本❶建立 Google Maps 实例❷，将初始位置❸显示在地图中央。然后代码获取 navigator 对象的 geolocation 对象并调用它的 getCurrentPosition 函数，传递的函数将来会以取得的位置值调用❹。如果请求成功，代码就会创建一个新的点对象❺，将地图居中显示在新的位置上❻，创建一个新标记❼，并在新位置上打开一个弹出窗口❽。

Geode 使用 Skyhook's Loki 服务。

3.4.4　原生浏览器支持

万维网联盟创建了 Geolocation API Specification，将其作为浏览器访问用户位置的标准方式。所有主流浏览器都支持这一规格。表 3-2 列举了当前主流浏览器支持地理定位 API 的情况。

比如你正访问一个位置感知网站，Firefox 3.5+就会询问你是否共享你的位置。如果同意，Firefox 就会收集附近的无线接入点以及你的电脑 IP 地址，然后将这些信息发送给默认的地理位置服务供应商 Google Location Services，从而估计你所在的位置。接着 Firefox 将估计出的位置分享给请求网站。如果你不同意的话，Firefox 就不执行任何操作。

表3-2　当前地理位置API规范的主流浏览器支持

Web浏览器	支持情况
Mozilla Firefox	3.5+
Chrome	5.0+
Opera	10.6+
Safari	5.0+
Internet Explorer	9.0+

使用以上列出的浏览器访问本地位置的实例代码与代码清单 3-8 中的相同。

3.5　小结

正如你在本章所看到的，平台、目标对象及位置，还有应用特性，决定了必须仔细挑选地图 API 和贴片供应商。

虽然出于整合的目的，所有 API 都支持 JavaScript，但 Flash 地图所提供的用户体验更流畅。

应该允许用户切换不同的地图视图，比如从卫星视图切换到地形视图，如果可能，允许用户改变地图贴片的样式。

确保目标位置的地图贴片是最新最精确的。如果覆盖的范围很广，那么你设计的应用就应该具有转换功能，以实现联机的不同地图贴片供应商之间的相互转换，这样才能为每个区域提供最好的地图贴片。

对移动 LBS 应用程序而言，选择特定的移动地图贴片会使小屏幕上的阅读更容易。

在所有平台上测试应用的性能，确保用户不会因为地图加载太慢或平移地图所花费的时间过长而放弃你的应用。毕竟，对他们而言换用另一个响应更快捷的 LBS 应用是在太容易了。

下一章，我们分析下一个对 LBS 应用很重要的因素：应用向用户展示什么，也就是内容。

第 4 章

内容选择

4

本章内容

- ❑ 从第三方获取内容许可
- ❑ 创建用户生成的内容
- ❑ 创建混搭网站

旅行者看到的是他的所见，而游客看到的是他想看见的。

——Gilbert keith Chesterton

随着越来越多的 Web 服务向移动服务靠拢，内容选项的幅度和范围也成倍增加，这就使应用开发者在给用户展示内容时有了更多选择。开发人员如果想吸引用户，就需要传送新鲜而刺激的内容，因为被吸引的用户是黏性用户。对基于位置的社交网络来说，这点尤其重要。

你可以在 LBS 应用中合并很多不同来源的数据。所有数据的主要特点是每个数据点都联系着一个位置，也就是说，每个点都有个经纬度。随着用户生成内容模式的出现，现在的 LBS 数据也能包含用户的评论、相片、视频及类似的东西，这样内容所有权问题就变得越来越模糊了。

另外，虽然多数用户生成的基于位置的数据来自于移动设备，但是允许用户在使用台式电脑时可以在线浏览自己及他人的数据也是非常重要的。

最后，为了成功，分析应用需求，并在服务中整合恰当的内容和与用户互动的模型是很重要的。还要明确该应用的服务条款，这样用户才能知道服务的具体限制条件和规定。图 4-1 显示了带有不同层面内容的 LBS 应用。

在这一章中，我们将讨论 LBS 应用中可以使用的内容，就从分析内容的许可及分发模式开始。

4.1 内容许可证

LBS 应用可用的每个内容都有与之相联系的许可证。内容许可证可以分为三类：商业的、免费使用的以及开源的。



图4-1　Centrl.com的饭店和折扣图层应用。带刀叉的图标代表折扣图层，其他图标代表饭店评价图层

- 商业的：你可以向 NAVTEQ 或 Tele Atlas 这样的公司付费，获得公司许可，在 LBS 应用中集成该公司各种各样的数据。比如你可以获取 NAVTEQ 提供的加油站价格数据许可，并在应用中使用。
- 免费的：根据服务条款（TOS），如果满足条件，有些商业数据供应商会允许你在应用中使用其数据而无须支付许可费用。举个例子，如果应用中有 Yelp 链接的话，Yelp TOS 就可以让第三方免费使用他们的数据。Yahoo!的 Upcoming 允许非营利项目免费使用其内容（对于商业项目，就必须与他们联系获得明确的许可）。
- 开源的：有些数据源，如 Wikipedia 和 OpenStreetMap，都是开放资源，大多数都以 Creative Commons License[1]（创意共享）的形式发布。任何人（营利的或非营利的）都可以获得允许使用其数据。

提示　THE OPENSTREETMAP LICENSE OpenStreetMap 数据的发布遵从开放内容许可，意在促进数据（既有商业的也有非商业的）的免费使用和重新分配。OSM 过去曾遵从 Creative Commons Attribution-ShareAlike 2.0[2]许可证，但他们在广泛的法律调查和社区咨询后，创建了一个新的许可证：从 Open Data Commons[3]（ODC）发展而来的 Open Database License（ODbL），这个许可更适合地图数据集。

LBS 应用的大多数内容是各种发布格式的数据源。

① http://creativecommons.org/
② http://creativecommons.org/
③ http://www.opendatacommons.org/

4.2　内容分发格式

LBS 应用最常见的分发格式是 GeoRSS、 GeoJSON 和 KML。每种格式都有其优势和劣势，下面让我们仔细地研究一下。

4.2.1　GeoRSS

GeoRSS①是新兴的将编码位置作为数据源的一种标准。它来源于 RSS，是博客和网站上最受欢迎的数据源和聚合格式。

迄今为止，GeoRSS 有两种主要的编码方式，叫作 GeoRSS 地理标记语言(GML)和 GeoRSS Simple。GeoRSS GML 是一个正式的 OGC GML Application Profile，比 GeoRSS Simple 支持的功能范围更广，值得注意的是除了使用 WGS84 经纬度之外，还使用坐标参考系统。另外还有一个较旧的并不鼓励使用的 W3C GeoRSS 序列化，不过它仍被广泛使用着。对于新的数据源来说，W3C GeoRSS 不再适用，但是对于现存的和较旧的数据源来说，还能用它进行处理。

代码清单 4-1 和代码清单 4-2 用两种不同的风格展示了 GeoRSS 格式数据。第一个代码清单展示了 GeoRSS 的简单格式，其简单序列化的目的就是在表示形式和概念上，最大限度地简化 GeoRSS。

代码清单 4-1　GeoRSS　Simple 数据源实例

```
<?xml version="1.0" encoding="utf-8"?>

 <feed xmlns="http://www.w3.org/2005/Atom"                    ❶ feed 标签的属性
       xmlns:georss="http://www.georss.org/georss">

   <title>Recommendations</title>
   <subtitle>Today's top recommendations</subtitle>           ❷ feed 标签包含其他标签
   <link href="http://centrl.com/"/>
   <updated>2008-12-13T18:30:02Z</updated>
   <author>
      <name>Centrl</name>
      <email>support@centrl.com</email>
   </author>
   <id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</id>

   <entry>      ❸ feed 标签也包含 entry 标签
                                                     ❹ 每个 entry 包含其他属性标签
      <title>Sushi Yasuda</title>
      <link href="http://centrl.com/location/ba44373ba0521e0e3cb7210"/>
      <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
      <updated>2008-08-17T07:02:32Z</updated>
      <summary>Great sushi</summary>
      <georss:point>40.714172,-74.006393</georss:point>
```

① http://georss.org/

```
  </entry>
 <entry>
    <title>Blue Ribbon Sushi</title>
    <link href="http://centrl.com/location/ff43473ba0521ade3cb1828"/>
    <id>urn:uuid:324356321-46fb8-4ebb-aaaa-80da344ef321</id>
    <updated>2009-08-17T07:02:32Z</updated>
    <summary>Festive atmosphere, great sushi</summary>
    <georss:point>40.714321,-74.12393</georss:point>
  </entry>
</feed>
```

这个实例展示了 GeoRSS Simple 格式的一些基本特征，比如如何指定 feed 标签的属性❶，每个信息源标签可以包含很多其他的属性标签❷，以及 entry 标签❸。每个 entry 标签又能有很多属性标签❹。

下面的代码实例展示了 W3C GeoRSS 格式。

4

代码清单 4-2　W3C　GeoRSS 信息源实例

```
<?xml version="1.0"?>
 <?xml-stylesheet href="/centrlstyle.xml" type="text/xsl" media="screen"?>


 <rss version="2.0"
      xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
      xmlns:dc="http://purl.org/dc/elements/1.1/">   ❶ rss 标签的属性

  <channel>                                            ❷ 在 rss 标签中，有
     <title>Recommendations from Centrl.com</title>       channel 标签及属性
     <description>Today's top recommendations</description>
     <link>http://centrl.com/</link>
     <dc:publisher>Centrl</dc:publisher>
     <pubDate>Thu, 27 Aug 2008 23:56:15 PST</pubDate>  ❸ channel 标签包含 item
                                                          标签
    <item>

         <pubDate> Thu, 27 Aug 2008 22:52:15 PST </pubDate>   ❹ item 标签包含的属性
         <title>Sushi Yasuda</title>
         <description>Great sushi</description>
         <link>http://centrl.com/location/ba44373ba05a</link>
         <geo:lat>40.714172</geo:lat>
         <geo:long>-74.006393</geo:long>
     </item>
     <item>
       <pubDate> Thu, 28 Aug 2008 22:52:15 PST </pubDate>
         <title>Blue Ribbon Sushi</title>
         <description>Festive atmosphere, excellent sake
           selection and great sushi</description>
         <link>http://centrl.com/location/ ff43473ba0521ade3cb1828</link>
         <geo:lat>40.714321</geo:lat>
         <geo:long>-74.12393</geo:long>
     </item>
   </channel>
   </rss>
```

这个例子指定了 rss 标签的属性❶。同时也告诉我们，feed 标签可以包含很多属性标签❷及 item 标签❸。而每个 item 标签各自又包含很多属性标签❹。

4.2.2　GeoJSON

GeoJSON①是基于 JSON（JavaScript Object Notation，JavaScript 对象表示法）的一种新的数据格式，可以表示各种地理特征包括点、线串、多边形 、多个多边形以及几何体集合。图 4-2 是 OpenLayers GeoJSON 工具②，它能简化多边形及各种图形的创建。

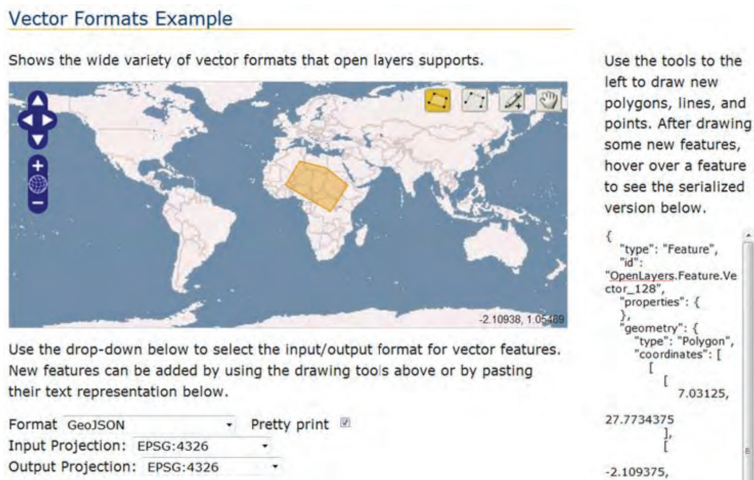


图4-2　利用 OpenLayers GeoJSON工具在地图上显示GeoJSON图形编码

GeoJSON 是比 GeoRSS 更加紧凑的数据格式（后者基于 XML）。当数据大小更重要时，选择 GeoJSON 就是一种好的方案。代码清单 4-3 是 GeoJSON 格式的一个实例。

代码清单 4-3　GeoJSON 数据源实例

```
{                          ◁——— JSON 完全采用 JavaScript 语法

   "type": "Feature",                ◁——❶ 键值对

   "geometry": {                        ◁——❷ 键对象对

      "type": "GeometryCollection",              ◁┐
      "geometries": [                             ❸ 包含键值对的几何对象

          {                                            ◁┐
            "type": "Point",                            ❹ 几何对象的混合数组
            "coordinates": [-74.006393,40.714172]
```

① http://geojson.org/
② mng.bz/8FW1

```
            },
            {
              "type": "LineString",
              "coordinates": [
                  [101.0, 0.0], [102.0, 1.0]
              ]
            }
        ]
    },
    "properties": {
        "name": "Sushi Yasuda",
        "category": "Restaurant",
        "link":"http://centrl.com/location/ba44373ba05a",
        "source":"http://centrl.com",
        "description":"Great Sushi"
    }
}
```

❺ 键值对集合

4

这个 JSON 实例（要注意的是由于 JSON 是纯粹的 JavaScript，这个例子可以用 `eval` 直接解析）展示了一组简单的键值对❶和键对象对❷，以及包含很多键值对和键对象对的几何对象❸。每个几何集合都是一系列几何对象的混合数组❹，每个对象都可以包含一批键值对❺。

4.2.3 KML

KML[①]（Keyhole 标记语言， Keyhole Markup Language）是采用 XML 语法的语言，用以描述地理注释和可视化元素适用于现存的或将来的在线二维地图以及三维 Earth 浏览器。KML 是专门为 Google Earth 开发的，Google Earth 最初叫做 Keyhole Earth Viewer，由 Keyhole 公司创建，该公司在 2004 年被 Google 收购。

现在，KML 是 OGC 的国际标准。其支持增长迅速，很多应用都可以显示 KML，包括 Google Earth、Google Maps、Google 手机地图、NASA World Wind、ESRI ArcGIS Explorer、Adobe Photoshop、AutoCAD 和 Yahoo! Pipes。

代码清单 4-4 展示了用 KML 数据将饭店显示为 Placemark 的实例。

代码清单 4-4　KML 信息源实例

```
<?xml version="1.0" encoding="UTF-8"?>

<kml xmlns="http://www.opengis.net/kml/2.2">
<Placemark>
  <name>Sushi Yasuda</name>
  <description>Great Sushi</description>
  <Point>
    <coordinates>-74.006393,40.714172,0</coordinates>
  </Point>
</Placemark>
<Placemark>
  <name>Blue Ribbon Sushi</name>
```

❶ Placemark 表示地物

❷ Placemark 标签包含属性和标签

❸ 经度、纬度和海拔

① http://www.opengeospatial.org/standards/kml/

```
    <description> Festive atmosphere, excellent
    sake selection and great sushi</description>
    <Point>
      <coordinates>-74.12393,40.714321,0</coordinates>
    </Point>
  </Placemark>
  </kml>
```

在这个实例中，可以看到一些带有属性和数值的基本 XML 标签，比如 `Placemark`❶。每个 `Placemark` 标签包含名称❷，以及说明和 `Point` 标签❸。

了解了 LBS 内容不同的格式后，我们将看一看怎么获取内容，可以是授权内容或让用户创建内容，或二者兼而有之。

4.3　第三方许可内容

可以放在地图上的任何内容都叫做兴趣点（POI）。顾名思义，兴趣点是人们认为可能有用或有趣的特定位置。很多第三方可以授权 POI 内容，其中很多供应商对 POI 进行了分类。有些 POI 聚合器给不同来源的 POI 数据提供一个简单的显示界面，比如 SimpleGeo 和 Factual。有些数据源经许可可以作为 POI 内容，这些数据源通常可分为以下几类。

- ❑ *进餐*——饭店、咖啡馆和快餐店。与这些资料相随的是用户评语和评级。例如 Citysearch.com、Yelp.com、Qype.com 和 11870.com①，如图 4-3 所示。
- ❑ *酒吧/俱乐部*——酒吧、俱乐部、休息室、冷饮店。这些资料也有用户评语和评级。比如 Citysearch.com、Yelp.com、Qype.com② 和 11870.com。
- ❑ *住宿*——酒店、汽车旅馆、度假村。比如 Cleartrip.com，如图 4-3 所示。
- ❑ *活动*——音乐会、会议、节庆。比如 Upcoming.org③、Eventful.com④和 Zvents.com⑤。
- ❑ *零售商店*——包括任何有零售店面的当地商铺，比如干洗店、杂货店或服装店。例如 Yahoo! Local⑥、Citysearch.com、MerchantCircle.com 和 Localeze.com。
- ❑ *服务项目*——任何提供服务项目的本地生意，如水管工和杂务工。例如 Yahoo! Local 和 MerchantCircle.com。
- ❑ *房地产*——这一类包括任何待售、出租的不动产或任何与不动产相关的信息，比如房产的历史价位信息。例如 the NYTimes Real Estate API、Zillow 和 Trulia。
- ❑ *气象*——地图上某个地方的天气状况和天气预报。例如 Weather.com 和 WeatherBug.com。
- ❑ *新闻*——某地的当地新闻。比如 Topix.com 和 Daylife.com。

① http://11870.com/api
② http://www.qype.co.uk/developers/api
③ http://upcoming.yahoo.com/services/api/
④ http://api.eventful.com/
⑤ mng.bz/3gs8
⑥ http://developer.yahoo.com/local/

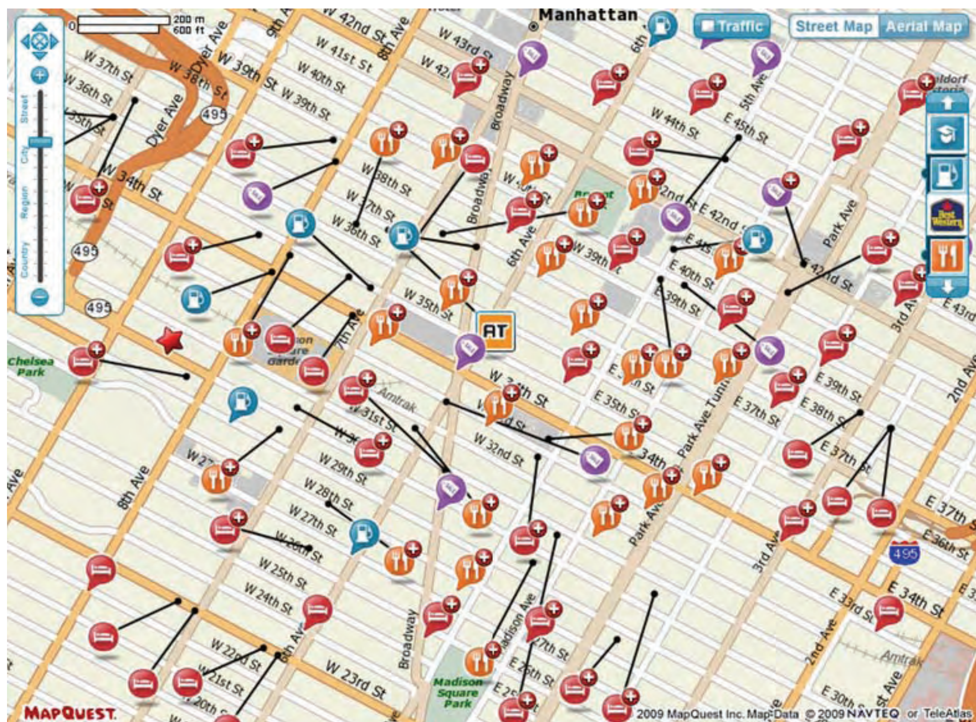


图4-3 显示了纽约市的饭店、酒店和加油站信息的MapQuest地图

- 照片和视频——地理编码照片和视频。比如 Flickr.com[1]、Panoramio.com 和 Picasa.com。
- 交通——主要公路的实时交通状况。比如 Google Traffic 和 NAVTEQ 的 Traffic.com。
- 旅行——像酒店、班机和汽车租赁的旅行信息。例如 Kayak.com。
- 朋友——朋友当前的位置。例如 Facebook、Twitter 和 Google Buzz。

另一个获取内容的方法是让用户来创建。多数应用通常会从第三方获取一些许可的内容，再让用户添加用户资料创建一套独特的数据。

4.4 用户生成的内容

第一代 LBS 应用多数是静态内容，就像是《米其林指南》(*Michelin Guide*[2])。第二代应用有了些由用户生成的与实际内容相关的反馈信息，像 Lonely Planet。到了现在，第三代 LBS 应用完全是用户生成内容（UGC），比如 Wikimapia 和 Virtual Tourist。用户在地图相应位置上添加评论、照片和视频资料，与其他人分享，如图 4-4 所示。

① http://flickr.com
② 法国出版的一份旅游饮食指南。——译者注

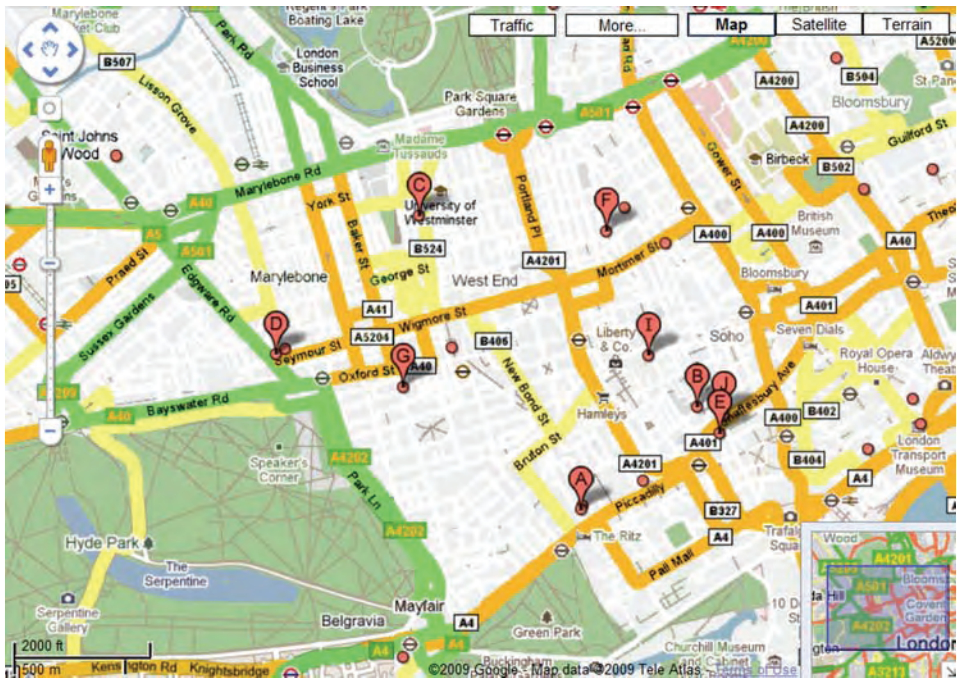


图4-4　Google Maps上显示的用户生成内容——伦敦海鲜饭店

UGC 是一个基本趋势，会开创 LBS 应用的未来。以鼓励用户参与，让用户生成内容的方式构建应用是很重要的。鼓励用户生成应用的方式很多，比如用户可以参与投票、为贡献内容排名、实现个人收藏功能以及与所有社会化媒体渠道共享。

由用户生成内容的应用提出了一些需要讨论的重要问题。

4.4.1　隐私

对于 UGC LBS 应用来说，隐私掌握在用户手中。用户可以选择做什么以及内容的隐私或公开程度。每个应用中都有明确的服务条款说明，这样用户在 Wikimapia[①]上添加条目时就会确切地知道其内容会被如何使用。很明显，如果添加的内容涉及隐私，就不应该添加到公开的数据库中。

4.4.2　质量

协作性的、开放的、允许任何人随意添加任何内容的应用，都存在准确、公平、质量，以及创造性工作及努力的可持续性问题。对使用 UGC 的 LBS 应用来说，这一点更为重要。原因在于 LBS 数据的准确性对应用来说是至关紧要的。没人希望开半个小的车找一个实际并不存在的饭

① http://wikimapia.org/terms_reference.html

店。LBS 应用必须非常谨慎地对待用户怎样添加内容，必须非常小心地通过协作和社区筛选来核实这些资料。

4.4.3 版权问题

如果用户添加的 UGC 受版权保护（比如相片或媒体文件），那么分清责任就是另一个重要的问题。众包、协作筛选以及非常清楚的服务条款都是避免这类问题的关键。

4.4.4 实施和性能

接下来的问题就是索引所有的空间 UGC 内容，反过来为用户提供实时的服务。必须充分优化数据库和应用界面，以便用户在特定领域内快速访问 UGC。对数据库来说，选择恰当的表布局以及正确的索引都很重要。

提示 POSTGIS PostGIS[①]为 Postgre-SQL[②]对象关系数据库添加了地理对象支持。实际上，PostgreSQL 服务可以使用 PostGIS 空间数据，可以将 PostGIS 作为后端空间数据库用在地理信息系统中，类似于 ESRI 的 SDE 或 Oracle 的 Spatial 扩展。PostGIS 遵循 OpenGIS[③] Simple Features Specification for SQL[④]，而且经验证与其 Types 和 Functions 相容。PostGIS 已公布遵循 the GNU General Public License（GNU 通用公共协议证书）。

知道了如何从不同输入源和用户那获取内容后，接下来我们要分析一下如何利用不同数据源快速创建有效的应用。

4.5 Mashups（混搭网站）

混搭网站（或混合 Web 应用）是合并多个外部来源的数据或功能，创建一种新服务的网页或 Web 应用。混搭网站能提供比单个服务更多的功能。使用开放的 API 和基于 Web 2.0 技术（比如 Ajax）的数据源，混搭网站常用很短的时间就能实现快速整合。图 4-5 展示了典型混搭网站的架构。

例如在地图上结合房地产数据和最近的犯罪信息，给买房人创建一个实用的服务，帮助他们在买房时选择最好的住宅区和房子。

① http://postgis.refractions.net/
② http://www.postgresql.org/
③ http://www.opengeospatial.org/
④ http://www.opengeospatial.org/standards/sfs

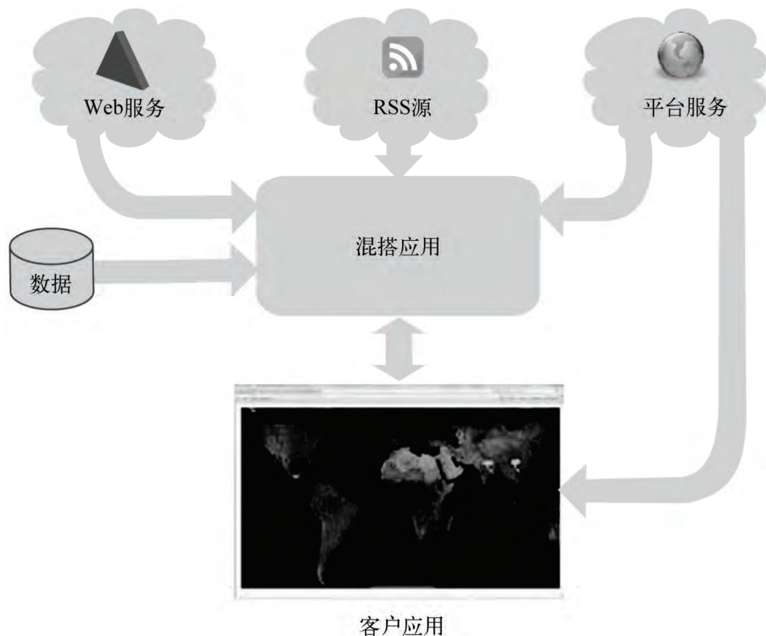


图4-5　一个典型混搭网站应用的架构。这个网站从不同来源获取信息，比如网络、RSS信息源和数据库，将这些信息整合在一起（来源：mng.bz/88yF）

由于采用公开的 API 和数据源，建立混搭网站所需的成本大大降低（接近为零），所以最近几年混搭网站变得非常受欢迎。流行网站 ProgrammableWeb.com[①]列举了由 1440 个 API 构成的 4298 个混搭网站。

现在了解一下利用这些数据源能创建什么样的混搭网站。

4.5.1　消费者混搭网站

消费者混搭网站旨在服务普通大众。例如 Intel 的 Mash Maker[②]能帮任何人创建混搭网站，如图 4-6 所示。Mash Maker 是一个浏览器插件，主要用于 Firefox，部分支持 Internet Explorer，就能让用户在现有网站上创建自己的混搭网站。

① http://programmableweb.com
② http://mashmaker.intel.com/web/index.php

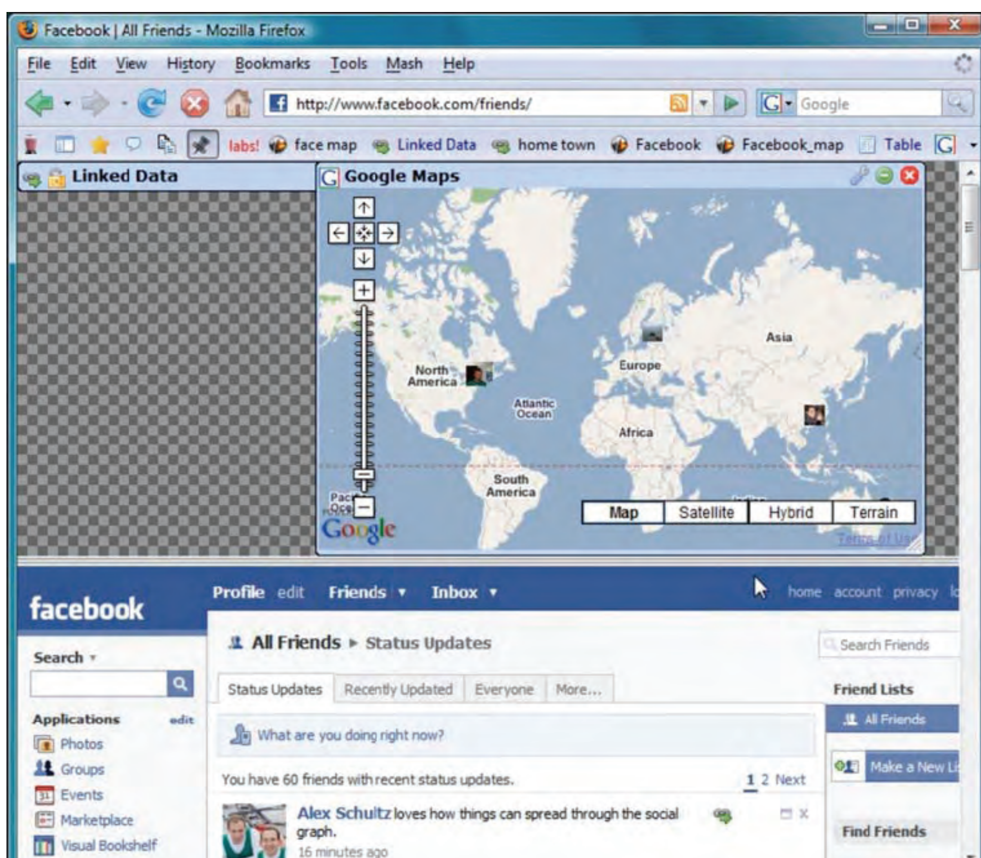


图4-6 用Intel Mash Maker混合Facebook和 Google Maps

4.5.2 数据混搭网站

数据混搭站点将来自多个地方的相似性的媒体和信息集中起来，通过统一的界面展示出来。一个例子就是 Havaria Information Services Alert-Map[1]，它将恶劣天气条件、生物危害以及地震信息相关的 200 多种数据源结合起来，并将其展示在世界地图上；另一个例子就是 Chicago Crime Map[2]，如图 4-7 所示，显示了芝加哥地区的犯罪率和犯罪地点。

① http://hisz.rsoe.hu/alertmap/index.php?lang=eng

② http://chicago.everyblock.com/crime/

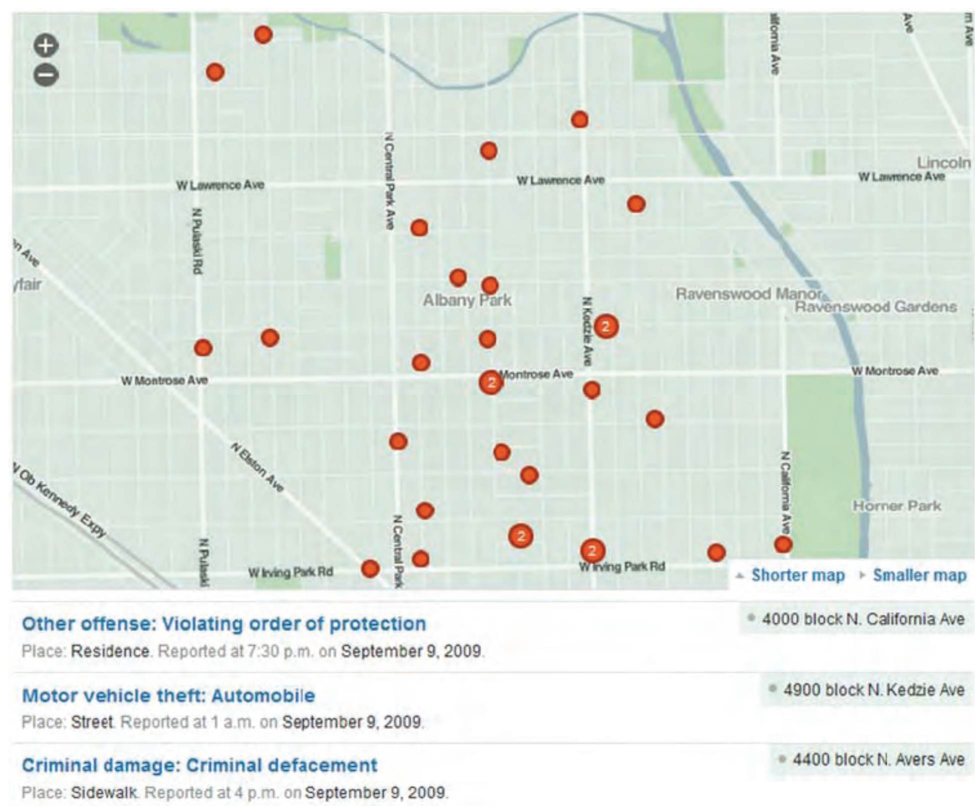


图4-7　芝加哥犯罪分布图

4.5.3　商业混搭网站

商业混搭站点把数据集中起来并以统一的界面显示，支持公司间的协作。不同的组织可以开放各种信息源，并将它们转换成新闻源或微件等形式的可混搭的资源。这些资源能在新应用中动态装配，以应对日常的商业挑战。这种混搭网站能够为商业用户减少事务积压、降低发展成本、提高现有资产的利用率，从而让更多人有效地消费。

例如，IBM Business Mashup Center[①]帮助企业方便快捷的创建商业混搭网站。

提示　YAHOO! PIPES　Yahoo! Pipes，如图 4-8 所示，是 Yahoo!的一个 Web 应用，能提供图形用户界面用以建立数据混搭网站。Yahoo! Pipes 支持聚合 Web 信息源、网页和其他服务，创建并公布基于 Web、不同来源的应用程序。利用这个应用，用户能输出不同来源的信息，而且能够建立内容修正的规则（例如，筛选）。

---

① mng.bz/ZdS5

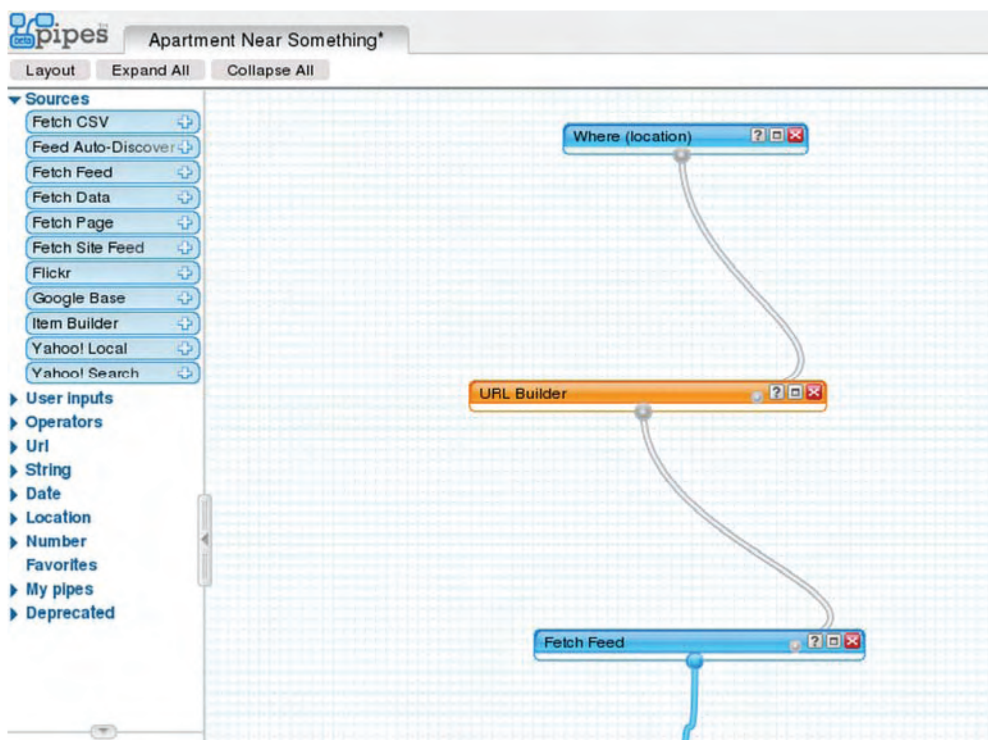


图4-8 Yahoo! Pipes

现在让我们看一些代码示例，更好的理解如何创建混搭网站。

4.6 混搭网站编码实例

创建一个混搭网站，将 Yelp[1]商业列表显示在 Google Maps 上[2]。为此，你需要从 http://www.yelp.com/developers 上获得一个 Yelp API 密钥，在 http://code.google.com/apis/maps/signup.html 上得到 Google Maps API 密钥。图 4-9 展示了完成后的混搭网站。

可以用两种不同的方法完成这个混搭网站。一个是利用 KML 格式化输入，另一个是直接用 JavaScript 语言通过 Google Maps API 加载 POI。第一项技术下一节会详细介绍。另一项技术会在 4.6.2 节讨论，即利用 JavaScript 从 JSON 信息源中加载各个地图项。

① http://www.yelp.com/developers

② http://code.google.com/apis/maps/signup.html

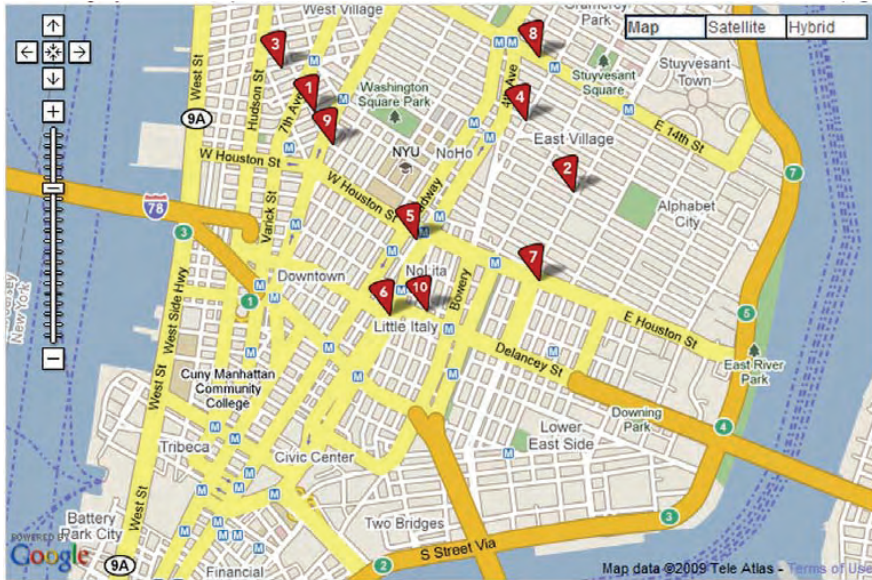


图4-9　Google Map上列举的Yelp门店

4.6.1　使用KML

如果你的 KML 文件中包含所有需要的 Yelp 商业列表，就可以利用 Google Maps API 对 KML 的支持，用两行代码搭建一个混搭网站，如下所示。

代码清单 4-5　利用 Google Maps API 中的 KML 支持

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
➥xmlns:v="urn:schemas-microsoft-com:vml">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8"/>

<script src="http://maps.google.com/maps?file=api&amp;v=2
    ➥&amp;sensor=false&amp;key=YOURKEYHERE" type="text/JavaScript">
    ➥</script>

<script type="text/JavaScript">
function initialize() {

    if (GBrowserIsCompatible()) {

        var map = new GMap2(
            ➥document.getElementById("map_canvas"));
```

包含 Google Maps 库

声明主脚本

❶ 检查 Goole Maps 的兼容性

❷ 创建 **map** 对象

```
        map.setCenter(
            ➥new GLatLng(37.4419, -122.1419), 13);        ❸ 设置地图中心点

        map.setUIToDefault();                               ❹ 设置默认 UI 选项

        var gx = new GGeoXml(
            ➥"http://example.com/my-data.kml");          ❺ 把 KML 数据导入对象

        map.addOverlay(gx);        ❻ 将 KML 对象放到地图上
    }
}
</script>
</head>

<body onload="initialize()" onunload="GUnload()">

    <div id="map_canvas"
      ➥style="width: 500px; height: 300px"></div>        声明容器 div
      </body>
</html>
```

这段代码先检查用户的浏览器是否与 Google Maps 兼容❶，然后创建了一个地图对象❷，将其居中显示❸，将 UI 设置为默认值❹。在赋完初值之后，将 KML 文件导入一个 KML 对象❺，并将其内容作为叠加层添加到地图上❻。叠加层是 Google Maps API 的结构，能在一个现有的地图上显示数据，该层能作为一个整体来操控。

4.6.2 使用JavaScript

如果没有 KML 文件，但又希望用户看地图时获取到 Yelp 的动态数据，就可以利用 JavaScript 解析输入的数据，在地图上为各个项目创建标记。下面展示了带有评论的混搭网站代码。

代码清单 4-6 Yelp/Google Maps 混搭网站的完整源代码实例

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/
     TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
    <title>Yelp Search API Example</title>

    <style type="text/css">                                声明 CSS 属性
        html, body {width: 100%; height: 100%; font-family: arial;}
      body {margin:0;padding 0;overflow: hidden;}
      #mapContainer {padding-top: 50px;}
      #map, #mapContainer {width:100%; height: 100%;}
      #top {position:absolute; top:0; left:0;
          ➥width: 100%; height: 50px; line-height: 50px;}
      #spinner { visibility: hidden; margin-left:3px;}
      #poweredby, #searchbox {line-height: 50px;}
      #searchbox {text-align: center;}
      #poweredby { float: right; margin-right: 3px;}
```

```
    #poweredby img { vertical-align: baseline;}
    .marker {font-size: 11px;}
    .marker .businessimage { float: left;}
    .marker .ratingsimage {vertical-align:middle; margin-top:0px;}
    .marker .businessinfo { margin-left: 110px;}
</style>

<script src="http://maps.google.com/maps?file=api
  ➥&amp;v=2&amp;key=[YOUR GOOGLE MAPS KEY]"          ❶ 包含 Google Maps 库
  ➥type="text/javascript"></script>

<script type="text/javascript">
 var YWSID = "[YOUR YWSID KEY]"; // 输入你的 API Key
                                          ❷ 将局部变量初始化为 null
 var map = null;
 var icon = null;
                                          ❸ 声明主脚本
function load() {                                         ❹ 创建地图对象
    map = new GMap2(document.getElementById("map"));
    GEvent.addListener(map, "load", function() {updateMap();});
    map.setCenter(new GLatLng(37.7916,-122.4418),13);
    map.addControl(new GLargeMapControl());               ❺ 居中地图
    map.addControl(new GMapTypeControl());
    map.setMapType(G_HYBRID_MAP);


    if (window.attachEvent) window.attachEvent(
        ➥"onresize", function() { map.checkResize()} )   ❻ 让地图随窗口缩放而变化
    else if (window.addEventListener)
        ➥window.addEventListener(
        ➥"resize", function() { map.checkResize()}, false);


    icon = new GIcon();                                   设置标记图标
    icon.image = "images/marker_star.png";
    icon.shadow = "images/marker_shadow.png";
    icon.iconSize = new GSize(20, 29);
    icon.shadowSize = new GSize(38, 29);
    icon.iconAnchor = new GPoint(15, 29);
    icon.infoWindowAnchor = new GPoint(15, 3);
}
                                          ❼ 为 API 请求构建 URL
function constructYelpURL() {
 var mapBounds = map.getBounds();
    var URL = "http://api.yelp.com/" +
        "business_review_search?"+
        "callback=" + "handleResults" +
        "&term=" + document.getElementById("term").value +
        "&num_biz_requested=10" +
        "&tl_lat=" + mapBounds.getSouthWest().lat() +
        "&tl_long=" + mapBounds.getSouthWest().lng() +
        "&br_lat=" + mapBounds.getNorthEast().lat() +
```

```
            "&br_long=" + mapBounds.getNorthEast().lng() +
            "&ywsid=" + YWSID;
        return encodeURI(URL);
    }

    function updateMap() {                    // 提交表单时调用
        document.getElementById(
        ➥"spinner").style.visibility = 'visible';      // ❽打开旋转动画

        var yelpRequestURL = constructYelpURL();      // 取得调用 Yelp 列表的 URL
        map.clearOverlays();

        var script = document.createElement('script');
        script.src = yelpRequestURL;
        script.type = 'text/javascript';
        var head = document.getElementsByTagName('head').item(0);
        head.appendChild(script);             // ❾请求 API
        return false;
    }

    function handleResults(data) {
        document.getElementById("spinner").style.visibility = 'hidden';
        if(data.message.text == "OK") {
           if (data.businesses.length == 0) {
                alert(
"Error: No businesses were found near that location");
                return;
            }

            for(var i=0; i<data.businesses.length; i++) {
              biz = data.businesses[i];
                createMarker(biz,
            ➥new GLatLng(biz.latitude, biz.longitude), i);      // 在地图上放置标记
            }
        }
        else {
            alert("Error: " + data.message.text);
        }
    }

    function generateInfoWindowHtml(biz) {    // 格式化并返回信息窗口的 HTML
     var text = '<div class="marker">';

        text += '<img class="businessimage" src="'+biz.photo_url+'"/>';

        text += '<div class="businessinfo">';
        text += '<a href="'+biz.url+'"
   target="_blank">'+biz.name+'</a><br/>';
      text += '<img class="ratingsimage"
   src="'+biz.rating_img_url_small
  ➥ +'"/> based on ';
      text += biz.review_count + ' reviews<br/><br />';
      text += formatCategories(biz.categories);
```

```
    if(biz.neighborhoods.length)
        text += formatNeighborhoods(biz.neighborhoods);
    text += biz.address1 + '<br/>';
    if(biz.address2.length)
        text += biz.address2+ '<br/>';
    text += biz.city + ', '
➥ + biz.state + ' ' + biz.zip
➥ + '<br/>';
    if(biz.phone.length)
        text += formatPhoneNumber(biz.phone);
    text += '<br/><a href="'+biz.url+'" target="_blank">
➥ Read the reviews</a><br/>';
    text += '</div></div>'
    return text;
}

function formatCategories(cats) {
 var s = 'Categories: ';
    for(var i=0; i<cats.length; i++) {
        s+= cats[i].name;
        if(i != cats.length-1) s += ', ';
    }
    s += '<br/>';
    return s;
}


function formatNeighborhoods(neighborhoods) {          格式化邻居的 HTML
 s = 'Neighborhoods: ';
    for(var i=0; i<neighborhoods.length; i++) {
        s += '<a href="' + neighborhoods[i].url
➥ + '" target="_blank">' + neighborhoods[i].name + '</a>';
        if (i != neighborhoods.length-1) s += ', ';
    }
    s += '<br/>';
    return s;
}

function formatPhoneNumber(num) {          格式化手机号码的 HTML
 if(num.length != 10) return '';
    return '(' + num.slice(0,3) + ') ' + num.slice(3,6)
➥ + '-' + num.slice(6,10) + '<br/>';
}

function createMarker(biz, point, markerNum) {          为指定门店创建标记
 var infoWindowHtml = generateInfoWindowHtml(biz)
    var marker = new GMarker(point, icon);
    map.addOverlay(marker);
        GEvent.addListener(marker, "click", function() {
            marker.openInfoWindowHtml(infoWindowHtml, {maxWidth:400});
        });
                                          ⑩ 打开第一个标记
        if (markerNum == 0)
```

```
      marker.openInfoWindowHtml(infoWindowHtml, {maxWidth:400});
    }

    //]]>
    </script>
  </head>

  <body onload="load()" onunload="GUnload()">
    <div id="top">
        <div id="poweredby">Powered by <a href="http://www.yelp.com">
     ➥<img src="http://static.px.yelp.com/i/map/miniMapLogo.png"
     ➥border="0" /></a></div>
        <div id="searchbox">
            <form>
                Search for <input type="text" id="term"
     ➥name="term" value="sushi"/>
     ➥<input type="button" value="Search"
     ➥onclick="return updateMap();"/>
                <img id="spinner" src="images/spinner.gif" />
                <span class="error" id="errorMessage" />
            </form>
        </div>
    </div>
    <div id="mapContainer"><div id="map"></div></div>
  </body>
</html>
```

这段代码包含了 Google Maps API 程序库❶，首先要初始化一些变量❷。然后加载主脚本❸，创建地图对象❹，并居中显示❺。如果用户调整浏览器窗口的大小，那么这段代码也一定会跟着调整地图的大小❻。

然后构建 URL 从 Yelp 取得数据❼。清除现有的叠加图层，调用 URL❾，提供一个加载的图标❽，然后开始处理返回的数据。检查返回的数据是否有效很重要，因为对世界上的某个区域来说，从 Yelp 那不可能获取到数据。

如果有列表从 Yelp 返回，在将每个字段做初始化后，再以标记形式添加到地图上。创建完所有标记后，就会自动打开第一个标记❿，这样，地图就可以移动到该位置以便向用户显示第一个列表。

4.7　小结

位置是我们日常所用数据中新的令人兴奋的一部分。所有手机很快都将包含位置功能，用户也会要求在所有应用中使用位置信息。因此在应用程序中整合内容与位置很重要，只有通过这种方式，用户才有可能更好地使用应用。

向 LBS 应用中添加内容，必须仔细了解内容的许可条款。

确定内容来源后，还要根据平台部署情况选择最好的分发格式。如果要通过一个不稳定的数据连接方式下载大量信息到手机上，就要选择简单的格式，比如 GeoJSON。在应用中，应该允许用户添加自己的评论、相片和视频文件，但是必须小心处理这些文件的准确、质量和隐私问题，

实现社区筛选功能。

如果应用中添加了很多内容来源，不要因为过多的标签而让地图显得过于拥挤。利用地图图层功能，用户就可以根据需要打开或关闭数据。

另外，所有内容（许可的或者用户产生的）本地化很重要。确保数据库中包含为每个 POI 设定相应地区的语言字段，通过各种办法将文本自动翻译成用户本地的语言。

下一章，我们将会讨论一些利用 LBS 的消费者应用。

Part 2 第二部分

技　　术

本部分主要介绍创建位置感知应用程序所需的技术。

定位应用涉及很多不同的技术，如移动平台、连接问题、后台服务、Web 浏览器中的应用程序、地图数据源、内容格式等。

第 5 章主要介绍不同类型的消费者应用，以及怎样将不同技术和内容组合起来创建令人兴奋的全新用户体验。

第 6 章介绍已有的主要移动应用开发平台，如 iPhone 和 Android。每个平台都举了很多例子，这样你就可以了解如何创建更出色的位置感知移动应用。

第 7 章主要讨论关于连接的问题。实时定位用户位置时，怎样才能确保应用程序能有最佳的表现？用户在两个小时内会把电量耗尽（并立刻卸载你的应用程序）吗？

第 8 章着眼于服务器端位置感知应用的集成，以及数据与应用最有效的连接方法。

掌握本部分的内容之后，你就能为自己的位置感知应用选择正确的技术了。

第 5 章

消费者应用

5

本章内容

- 目的地导航应用
- 与他人或当地保持联系的应用
- 提供娱乐的应用
- 前沿应用

最近，移动消费者可使用的 LBS 服务数量和种类都在迅速增加，其中包括实用服务和休闲服务。无论是 iPhone 还是高端手机，LBS 应用都受到消费者的青睐，其手机下载量占到了市场份额的大约 25%（正如第 1 章提到的，这个百分比还在上升）。

说到实用的应用，导航应用和找餐馆等可下载的小工具都算，它们需要连接到中央服务器以获取用户的实时位置信息，为移动设备创造了全新体验（并且快速渗透到便携式导航设备市场）。

另一方面，现在的移动社交网络（MSN）超越了仅仅关注在线和上下文（我是否连接网络了？我在干什么？）的局限，开始关注位置（我在哪？我的朋友们在哪？）。

本章中，我们将展示现有的各种不同的应用，弄清楚不同领域可能存在的巨大机会，满足所谓的长尾移动消费需求。长尾理论表明，如果分销渠道足够庞大，通过满足利基消费者市场的新需求，建立一个能够独立发展并成功的业务是很有可能的。今天，应用商店的全球用户达到了数十亿，新应用所能服务的对象比以往任何时候都要多。

移动用户的需求主要分布在三个领域：

- 目标导航
- 与他人或当地保持联系
- 娱乐或游戏

针对每一个需求，我们都将讨论开发者、初创企业以及主流玩家是如何尝试弥补市场空白的，并通过一些真实的案例阐述。本章还会介绍较新的增强现实功能。这是一个开发者正在努力奋斗的前沿领域，很多令移动用户从未想见的应用正在开发中。

5.1　目标导航

早期的手持或车载轻型导航设备、个人 GPS 辅助导航的成本只有少数富人可以承担。今天，具有导航或位置识别（通过设备的硬件或软件）功能的移动设备大约占手机数量的一半。

手机功能的增强，为开发定位应用提供了广阔空间。

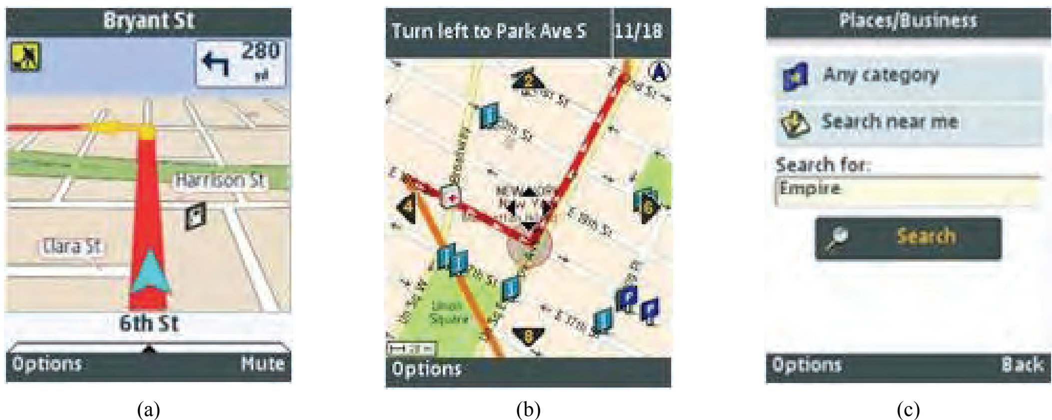
辅助导航设备利用手机的地图和定位功能，指引用户走到或开车到目的地。这些设备解决了从 A 点到 B 点的基本问题。

在本节中，我们将讨论 Telmap 和 Nokia 公司手机导航方案的一些实例。这些方案是他们为满足移动卫星导航需求而开发出来的。Ovi Maps 对一些 Nokia 智能手机用户提供的服务是免费的，而 Telmap 则提供收费服务。

为用户查找饭馆或加油站等兴趣点，是导航应用越来越受欢迎的原因。其中一些比较成功的实用（简单的）应用的分发模式已经遍及开发平台。一些服务提供商已经致力于创建基础模块，然后授权第三方来构建他们自己的应用，而不仅仅是提供应用服务。uLocate 现在仍是由公司来提供完备的位置平台，在此平台上开发应用程序（虽然他们现今的商业模式已有所改变；毕竟是移动行业，发展太快！）的最好案例。我们将通过了解 uLocate 如何成功吸引开发人员在他们的 WHERE 平台上开发应用程序来结束本节。

5.1.1　智能手机上的GPS辅助导航Telmap

Telmap 称自己是为所有主流移动平台提供的完整 GPS 导航解决方案。它可以从行人导航转换到车载导航。在为行人导航时，用户可以通过点击地标来获取更多的信息；在为汽车导航时，能根据实时交通状况，及时为用户提供有语音提示的最佳行车路线（如图 5-1 所示）。



(a)　(b)　(c)

图5-1　Telmap的移动解决方案

多种不同的智能手机每月花费 4 美元就可以下载 Telmap 应用，但它并不适用于没有 GPS 功能的中低端手机。对于大多数智能手机来说，运行 Telmap 应用的同时仍然可以接打电话。一些

批评者指出，手机屏幕太小，无法实现完美导航，然而随着新款智能手机的屏幕越来越大，这已不是个问题了。

Telmap 应用允许用户查询信息并将查询结果通过短信发送给朋友。为应对包含定位搜索功能的免费社交应用的潜在威胁，Telmap 正着重在自己的应用上构建社交网络元素。虽然这个策略是产品扩展的必然结果，但是鉴于消费者经常不太情愿使用这种混合了实用和休闲元素的应用，因此该策略能否成功，还有待观察。

可以肯定的一点是，因为溢价定位 Telmap 从一开始就持续盈利，而且 Telmap 及其竞争对手证明，利用移动 LBS 可以建立一个成功的可持续的业务（尽管商业模式变化迅速，见 5.1.4 节）。

图 5-1 显示的是 Telmap 移动解决方案的截屏画面，从左至右依次是车载导航（a）、行人导航（b）和搜索当地企业的功能（c）。了解了 Telmap 的收费服务后，我们来看一下来自 Nokia 的 Ovi Maps。

5.1.2　基于Nokia手机的GPS辅助导航Ovi Maps

大多数 Nokia 手机上都预装了 Nokia 地图（现称为 Ovi 地图，与 Nokia 旗舰应用商店一致），能在 NAVTEQ 地图上高分辨率显示众多不同的兴趣点（POI），比如药店和加油站。

Nokia 最初在免费版本上提供一些收费服务，包括交通信息提示服务、驾驶服务（汽车导航）和 City Explorer（行人导航）服务。City Explorer 服务反映出了 Nokia 的战略方向，2008 年 Nokia 在全球移动大会上宣布它会成为全球移动定位服务的领导者。在移动定位服务还没有被广泛采用、只不过是这一策略发展的一小部分时，Nokia 就推出了行人智能导航服务，并最终成功收购了美国的地图巨头 NAVTEQ。

行人智能导航服务允许客户以 3D 的方式查看地标（如图 5-2 所示），并通过 Lonely Planet 和 ViaMichelin 收费信息给用户提供丰富的使用体验。尽管每年 4.5 欧元的费用是很多人都可以承受的，但 Nokia 仍然要和其他很多免费的服务（包括 Google Maps 的移动服务，我们将在 5.1.4 节详细讨论）进行激烈地竞争。

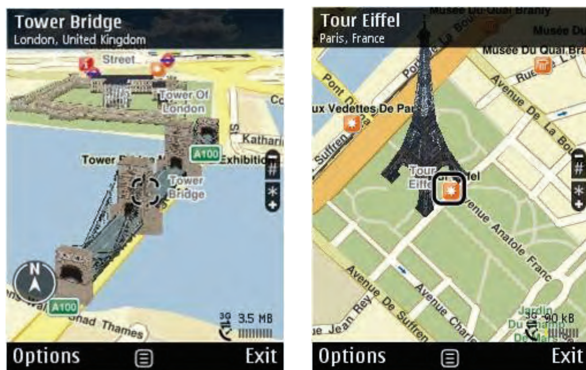


图5-2　在手机屏幕上显示的由Ovi Maps的City Explorer服务生成的伦敦塔桥和巴黎埃菲尔铁塔的高分辨率3D图像

驾驶服务（如图 5-3 所示）属于完善的卫星导航的一部分，最初在很长一段时间内它持续获利。

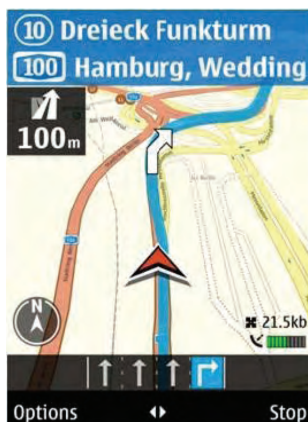


图5-3 Ovi Maps利用GPS通过手机屏幕进行实时智能汽车导航

5

2009 年 5 月，Nokia 宣布推出 Ovi Maps Player 应用程序接口，允许第三方将 Ovi 地图嵌入网站。这种在网站和手机之间同步的能力使Nokia地图比移动定位设备市场中的竞争对手更有优势。不久之后，Google 于 2009 年 10 月宣布将提供免费的移动导航服务（见 5.1.4 节），这一惊人的消息令 Nokia 决定为 10 款 Nokia 手机（包括 Nokia N97 mini、Nokia 5800 XpressMusic 和 Nokia E72）提供免费的智能导航和城市向导服务。Nokia 的计划表明它在未来将把免费服务推广到所有 Nokia 手机上。

了解了两家向终端用户提供移动导航服务的公司的例子后，我们将讨论一个向移动运营商销售定位平台的公司的例子。为此目的，该公司的发展重点就是培育开发人员的生态系统。接下来，我们将看看他们是怎么做到的。

5.1.3 创建导航部件平台uLocate

我们在第 1 章提到，uLocate 为第三方 LBS 应用开发了一个 WHERE 创新平台。它不仅能够导航，还能搜寻位置。

通过向外部开发人员开放平台，uLocate 鼓励其团队创建了很多的部件。这些部件都在它的部件库中，如图 5-4 所示。uLocate 声称利用他们公司现成的 XML 标签和平台，在 48 小时内就可以构建出一个小部件。这些小部件包括：

- 地震探测器
- 汽车租赁
- 啤酒厂探测器
- 天气状况

uLocate 就是用这方法确立了该公司在美国位置服务平台中的领先地位，并鼓励不断推出新应用。eBay 在 2011 年 4 月以 7500 万美元收购了 uLocate（现改名为 WHERE），这也证实了 uLocate 的成功。



图5-4　WHERE部件库，开发者可以在这里发布其利用uLocate移动定位平台开发的部件

为 iPhone 平台设计的交通应用软件（如图 5-5 所示）包含一些增值服务，如：

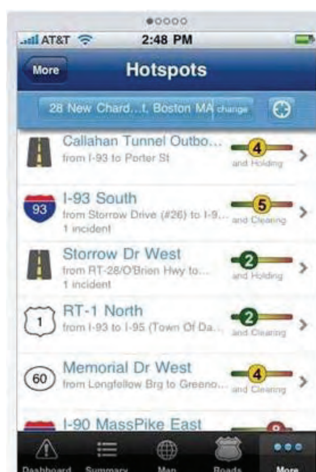


图5-5　在iPhone上运行的uLocate 的交通应用截屏，显示的是在一条通往马萨诸塞州波士顿市某地的典型线路上的交通热点

- ❑ 个人驾驶与 Traffic.com 网站同步，从而使用户能自定义路线
- ❑ 用实时和历史车流速度数据计算堵车的严重程度，并迅速通知司机
- ❑ 通过 NAVTEQ Map TP 平台，在移动地图上用不同颜色实时表示交通流量，并可以平移和缩放，以快速查看一个地区的交通阻塞情况，或是将 GPS 模式设置为跟随通勤者移动
- ❑ 综合分析包括事故、道路施工、重大事件、天气和交通阻塞情况在内的数据
- ❑ 城市里或所选路线中的交通热点

不难看出，这种应用将与现有的导航供应商，如 Telmap 和 Ovi Maps 进行正面竞争，而且为最终用户提供增值服务将对保持或获得市场份额非常重要。

这也是 Google 突然决定为它运行 Android 系统的设备提供免费地图导航的原因。接下来我们将讨论它是如何实现的，以及对竞争者产生了哪些影响。

5.1.4 Google的（免费）地图导航

2009 年 10 月，Google 宣布推出手机免费导航服务，这种服务将提供线路指示、实时路况信息更新和语音指令识别功能（如图 5-6 所示），此举震惊了市场。最初这种服务只适用于一种 Motorola Droid 新手机，并且仅在一个国家（美国）使用，而 Ovi Maps 已能在 74 个国家使用了，但是 Google 已经计划将此服务拓展到更多的手机上。



图5-6 基于Android的Google地图导航应用截屏，目前仅在美国Motorola Droid等少数手机上免费应用

Google 此举给移动产业带来很大的震动，导致卫星导航的先驱 TomTom 公司的股价在一天内就下跌了 20%还多。尽管 TomTom 在 2009 年 8 月向 iPhone 提供售价 100 美元的导航应用时取得了显著成功。

虽然从表面上看来，Google 给前面讨论的 TeleNav 等公司的收费卫星导航服务带来了负面影响，但事实上 Google 的举动证实了手机比单独的 GPS 更适合作为卫星导航的平台。然而，通过

实施标准的广告赞助商业模式，Google 很可能迫使竞争对手也遵循这种模式。因此开发者应注意开发有利于获得广告收入的应用功能和用户界面。

在社交网络领域中，发展了很多关于消费者 LBS 的初创企业和新应用，大小玩家们都渴望将网络领域中的这种爆炸式的增长或成功复制到手机上来。接下来，将研究新推出的 5 项服务的实例，对消费者来说，其中的每一项都有着独特的吸引力。

5.2　与他人或当地联系

GPS 和其他技术的推广使得社交网络能够添加定位功能，比如找朋友服务，这使人们更容易保持联系。

个人区域网的定义：基于位置的移动社交网络与一般的社交网络不同，它能够确定用户的位置。一些观察员借用 IT 产业的专业术语把这些新型社交网络称作个人区域网。

表 5-7 中比较了 iPhone 上一些著名的基于位置的移动社交网络的主要特征。我们已经在 5.1.3 节里对 uLocate 有了深入了解，并将在 5.2.1 节到 5.2.5 节中研究 Whrrl 和 Loopt（以及其他应用）的差异。

	Loopt	Moximity	Pelago (Whirrl)	Limbo	uLocate (Where)
Cost	Free	Free	Free	Free	Free
Location Availability	Everywhere	Austin, TX	Everywhere	Everywhere	Everywhere
Funding	$13.3M	Angel	$22.4M	$15M	$15.5M
Headquarters	Mountain View, CA	Austin, TX	Seattle, WA	Burlingame, CA	Boston, MA
Messaging	Yes	No	No	Yes	Yes
Map Friends	Yes	No	Yes	Yes	Yes
Friends' Exact Location	Yes	Kinda (Place)	Yes (?)	No (City, miles away)	Yes
Places/Events	Yes (Yelp)	Yes	Yes	Yes	Yes (Yelp, Others)
Place Types	All types	Restaurants, Bars, Special Events	All types	Food, Bars, Shops, Coffee, Fast Food, Gas, Night Clubs	Yelp, Starbucks, Zipcar, Gas Stations, Events
Place Reviews	Yes (Yelp, "Comments")	No	Yes	No	Yes (Yelp)
User Categorization	Friends, Members	Friends	Friends, Fans, Members	Favs, Friends, Contacts, Members	Buddies
Location Updating	Automatic	Manual	Automatic	Manual	Manual
Privacy Settings	Block Users, Report Abuse, Visibility Levels	None	Per-Friend Visibility Settings	Visibility Levels	Show or Hide Location
Import/Sync Contacts	Yes	Facebook and Twitter	Yes (only via email)	Yes (phone)	Yes (Facebook)
Share Photos	Yes	No	No	No	No
Advertisements	No	Yes	No	No	No
Wall Posts	No	No	No	No	No
Request Contact Info	No	No	No	No	No
Profile	Portrait, Gender, Looking For	?	Portrait, Name	Portrait, Age, Name, City, Country, Activity	Name, Status
Status Messages	Yes (w/photos)	Yes (Synced with Face-book, Twitter)	No	Yes (predefined Activity)	Yes
Web Hooks	Twitter, Facebook	Facebook	Facebook	No	Facebook, Twitter

图5-7　iPhone上比较流行的基于位置的移动社交网络的对比（来源：TechCrunch, 2008）

这些可接入移动平台的位置感知社交网络对移动和网络平台都造成了很大的影响，也给此新市场中那些渴望成功的新公司带来很大的动力。虽然最初只有初创企业在此领域寻求发展，但现在较大的社交网络和网络社会媒体大鳄们也开始涌进来。2009 年年初谷歌宣布推出其移动应用杀手锏 Google Maps 的 Google Latitude 功能，引起了很大震动。此服务推出后，在手机（由于很多手机上都预安装了 Google Maps，因此这样的手机占有的市场比例很大）上安装了 Google Maps

的移动用户就能在 Google Maps 上看到他们朋友的位置了（如图 5-8 所示）。



图5-8　Google Latitude屏幕截图显示了纽约市曼哈顿中心区三名在线成员的位置

尽管一些较小的 LBS 公司谴责 Google，认为此项服务的推出会给 Google 带来竞争优势，而这会给他们造成不利的影响，但事实上该服务并没得到广泛推广。该服务主要的功能是使朋友和家庭成员相互可见，并没有明显的切实的附加利益，而且隐私问题似乎造成公众没有广泛接受它。

其他社交网络巨头也在其移动服务中引进了定位元素，比如 Twitter 先是支持从其用户的推文时间线中获取位置，随后又开发了 Twitter Geolocation API，从而对其用户实现更精确的定位。

现在，Facebook、Twitter 和其他主要的在线社交网络已在他们的平台上添加了定位功能。（Facebook 于 2010 年年中在美国推出了 Facebook Places，并于 2010 年 12 月在欧洲推出该功能。）Google 在 2010 年年初进军社交网络领域，推出了 Buzz，毫无疑问，此举引起其他公司的争相模仿。

接下来，我们将会介绍一些更有趣的基于位置的社交网络，这些服务最初出现在新兴市场，在后来过去的两到三年内，就被推广成为一种大众服务了。另外，在本节最后，我们还会了解通过当地促销或打折活动，比如 Groupon 抵用券，将消费者与当地企业连接起来的新兴趋势。

5.2.1　用有地理标记的相片编故事的Whrrl

Whrrl 在基于位置的社交网络这一块，主要侧重于用户在特定地点拍摄并上传的照片。Whrrl 具有方便浏览的 iPhone 用户界面（如图 5-9 所示），它源源不断地将照片与用户评注关联起来，创造出类似于现场直播的故事场面。

Whrrl（由前 Amazon 员工创建）本来打算在 2007 年推出位置搜寻功能，但是后来主打照片故事了。

与 Facebook 和 Twitter（二者在社交领域都几乎是必不可少的）整合后，用户可以自己决定

邀请谁来浏览其故事。隐私问题也得到了充分的重视，对如何讲故事至关重要：只有信任的朋友才能访问用户的所有信息。



图5-9　Whrrl的iPhone应用截图，描述了由其中一位用户提供的个人相片故事，发生在附近的公共事件

该应用整合的另一个实用功能就是“最喜爱的地方”，这样用户就可以从他去过的地方清单中进行挑选。

Whrrl 的独特之处是能将用户的故事变成一种集体移动期刊，人们对同样的经历可以发表不同的观点，很适合会议或聚会等场合。这些故事也有自己唯一的 URL，这样在 Whrrl 之外也可以访问，甚至能由 Whrrl 网站（如图 5-10 所示）上的所有参与者编辑。



图5-10　Whrrl网站的主页，所有用户都可以浏览由带地理标签的照片（由Whrrl社区的会员拍摄）组成的公共故事，也能与故事位置相近的其他成员联系起来

2010 年 Whrrl 再一次改变了观念（证明应用开发需要不断地适应快速变化的消费者需求），现在允许用户在任意地点签到，从而发现志趣相投的人。Whrrl（通过它的总公司 Pelago）在 2011 年 4 月被 Groupon 收购，该公司计划停止 Whrrl 服务，但是会将其技术整合到自己的产品当中。

我们已经了解了基于位置的移动社交网络 Whrrl，接下来了解一下早期的 LBS 先驱——Loopt。

5.2.2　让位置进入主流的Loopt

可以把 Loopt 定义为一种移动社交地图应用，用户可以用它搜寻朋友和地点，其关键用户界面是一张地图，如图 5-11 所示。Loopt 利用位置数据（来自设备本身或运营商网络）显示用户朋友的位置及其状态（在线、离开及其他）。用户可以在一定距离内向朋友发送警报要求，并为其具体位置设置电子标签。



图5-11　Loopt移动应用截屏，显示了用户在地图上的位置、在线好友以及Loopt社区最近在此区域内留下的信息

Loopt 宣称它在美国（其运营的唯一国家）的用户已逾百万，其中多数是二十五六的年轻人，它能在大多数移动平台上使用，包括 iPhone、Blackberry 和 Motorola 及其他制造商生产的非智能手机。

Loopt 发展计划中的一个关键战略要素是要适用于众多移动运营商的平台，这样美国的每个主要移动运营商都能提供 Loopt 服务了。

Loopt 战略中的另一关键要素就是建立以会费为基础（带少量的广告）的商业模式。先为会员提供一个月的免费服务，随后每月收取 3 到 4 美元的费用。随着其他服务免费使用后，这种收费模式也随之改变，现在 Loopt 的服务是免费的了（带有广告）。

由于具有先发优势和大量的财政支持（至今已超过 1300 万美元），如今 Loopt 已成为该领域中最有名的 LBS 之一了。2008 年，Loopt 赞助了 *The Middle* 节目，成为首个也是唯一一个开展电

视宣传活动的 LBS。

Whrrl 和 Loopt 的主要市场都在美国国内，而 GyPSii 则始终如一地在全球提供服务。接下来我们将会了解 GyPSii 服务的一些主要特征。

5.2.3　让LBS服务全球的GyPSii

GyPSii 是得到最大资金支持的非基于美国的 LBS 初创企业之一，自从推出之后，已收到投资人逾 1300 万美元的投资。从一开始，它就提供完备的基于位置的社交网络应用，能够运行于 Symbian 和 iPhone 众多平台之上（如图 5-12 所示）。

GyPSii 是一个以地图为中心的应用，用户可以用它找到自己所在的位置、寻找朋友或地点，还可以为其他人标记电子标签和图像。这个技术完备的应用的主要缺点或许就是使用笨拙，要使用其功能，用户需要打开一系列的菜单和子菜单。实际上，它主要是在亚洲使用，亚洲人习惯于在移动应用上浏览更多的详细信息。虽然此公司并不公布会员号或故障，但是社区的亚洲特色对多数欲加入的成员来说是显而易见的。尤其在中国，GyPSii 已经适应当地的市场需求（以及关于 GPS 使用的法律规定），并大力推广其基于 Java 的应用。



图5-12　GyPSii最新的iPhone主屏幕显示了荷兰阿姆斯特丹市中心区（左），其places屏幕显示了按照用户的距离和更新时间列出的POIs

GyPSii 为移动开发人员提供了两个有用的启示：首先，精心设计且操作简单的应用程序，比过分修饰功能全面的应用程序更具吸引力；其次，对某些人群来说，应用程序的固有设计或外观或多或少有吸引力。在当今全球经济的形势下，这就意味着需要考虑不同大陆人们的喜好和品位，也要考虑通过本地市场调查来获取信息。

一些最初做位置感知应用开发的人员认识到，影响应用接受度的一个障碍就是服务使用的复

杂性。有个公司的目标就是开发易于使用的应用，这个公司就叫 Sniff。

5.2.4　使LBS更容易的Sniff

虽然 Sniff 的核心服务与其他移动社交产品相比并不独特，但是它却带来了这些服务收费机制的革新。它为一个简单的操作引进了定价方法：通过发送 SMS 寻找朋友或者查明他们所在的位置。

Sniff 最初在英国和斯堪的纳维亚（半岛）推出，后来也在美国推出了。从一开始（如图 5-13 所示），Sniff 就结合了 Facebook 的 Web 应用，因此从推出之日起，它就是一个比较新颖的应用。

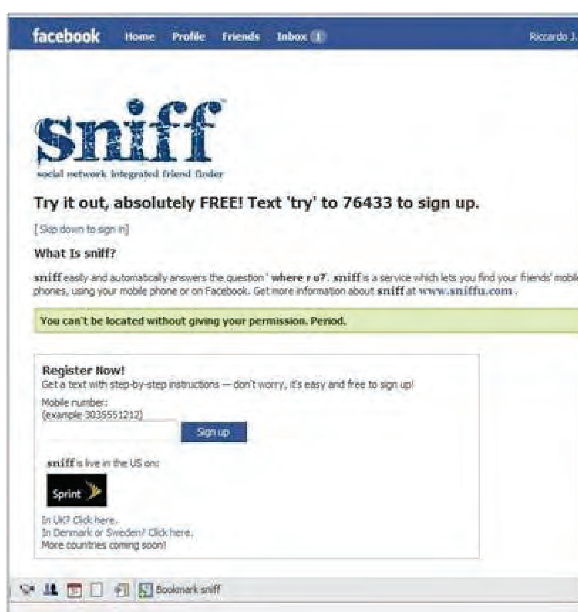


图5-13　Sniff以结合Facebook作为其服务的核心，引领了发展方向

Sniff 管理隐私的方法就是请求每一个成员允许他们的朋友来搜寻他们的位置。成员也能通过发送 SMS 把状态设置成不可见的。Sniff 最初的新颖性加上没有限制特殊手机（因为它使用的位置信息是从蜂窝网运营商那获取的）造成它非常受人关注。但是随着可以提供相似功能的免费服务的出现，人们对 Sniff 的热情锐减，后来这种服务就停止运营了。

5.2.5　通过LBS贯穿真实和虚拟世界的Friendticker

德国柏林的创业公司 Friendticker，出人意料地使用了从 LBS 应用取得的精确位置信息，并没有依赖不同定位技术的组合（如图 5-14 所示）。



图5-14　Friendticker允许其社区成员以精确的位置签到，这样在城市中任何地方，他们的位置精度都在1m以上。Friendticker将这种新概念称为超级定位

Friendticker 在某种程度上是通过服务标签或显示在商店或饭店窗口上的位置便签（如图 5-15 所示）来消除现实世界和虚拟世界之间的隔阂。这些便签能让用户通过 SMS、对条形码进行扫描，或者利用非接触式近场通信（NFC）技术（NFC 技术背后的 RFID 技术在第 1 章介绍过，当时将其与 GPS 技术作了比较）就可以签到某个位置。每一个便签都有唯一的号码来标识位置。这种方法能具有 1m 以内的定位精度，这显然是其他方法达不到的！读取 NFC 标签的客户端代码（用 Java ME 编写）示例如代码清单 5-1 所示①。



图5-15　Friendticker的标签不仅具有NFC功能，而且合并了QR（Quick Response，快速响应）代码，这样就能通过手机上的摄像头记录用户的位置。另外，用户可以利用标签上唯一的号码发送SMS。每个标签都附加了用户位置，具有唯一的数字识别号

Friendticker 将这种精确的定位方法称为超级定位（hyperlocalization）；要理解这个概念，可以用三组坐在相邻会议室的同事作为例子。从标准定位技术的角度看，他们处于相同的地方，

① 源自 2010 年 6 月的 Forum Nokia，www.wiki.forum.nokia.com，由 Gerald Madlmayr 贴出。

不能互动。然而利用 hyperlocalization，当一组同事腾出邻近的会议室时他们就能知道。同样，一个多层楼的大停车场，或其他人口密集的环境（比较典型的是在市中心），都可以找到类似的场景。

Friendticker 就这样使社交网络和有精确位置的现实世界产生互动了。与其他我们早先了解的应用一样，Friendticker 也能通过一个桌面应用与 Facebook 灵活地整合，提供实时推送的位置信息。

代码清单 5-1　读取外部 NFC 标签 UID 的 JME/Java MIDlet 代码

```
package at.nfcresearch.wima.examples;

                                                                      包含无接触通信包
import javax.microedition.contactless.ContactlessException;  <-
import javax.microedition.contactless.DiscoveryManager;
import javax.microedition.contactless.TargetProperties;
import javax.microedition.contactless.TargetListener;
import javax.microedition.contactless.TargetType;

                                                               包含 GUI 包
import javax.microedition.lcdui.Alert;
import javax.microedition.lcdui.AlertType;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Form;

import javax.microedition.midlet.*;

public class UIDReader extends MIDlet implements TargetListener,  <-
 CommandListener {                                                 ❶ 主类

    private Command exitCommand;
    private Form form;

    public UIDReader() {


        exitCommand = new Command("Exit", Command.EXIT, 1);      ❷ 创建 GUI 元素
        form = new Form("NFC-Research.at: UID Reader");  <-
                                                   ❸ 添加命令

        form.addCommand(exitCommand);  <-
        form.append("Touch Tag to read ID.");
        form.setCommandListener(this);

                                                  创建 Discovery Manager ❹

        try {
            DiscoveryManager dm = DiscoveryManager.getInstance();  <-
            dm.addTargetListener(this, TargetType.NDEF_TAG);
```

```
        } catch (ContactlessException ce) {
            displayAlert("Unable to register TargetListener: "
            +    ce.toString(), AlertType.ERROR);
        }

    }

    public void startApp() {
        Display.getDisplay(this).setCurrent(form);
    }

    public void pauseApp() {
    }

    public void destroyApp(boolean unconditional) {
    }

    /**
     * Implementation of the Call-Back Function of the TargetListener
     * @param targetProperties: Array of Targets found by the Phone
     *
     */
    public void targetDetected(TargetProperties[] targetProperties) {
                                                        ❺ 没找到标签就退出

        if (targetProperties.length == 0) {        <-
            return;
        }
                                                                ❻ 显示发现的
                                                                  第一个标签
                                                                  的 UID
        TargetProperties tmp = targetProperties[0];
        displayAlert("UID read: " + tmp.getUid(), AlertType.INFO);  <-
    }

    public void commandAction(Command command, Displayable displayable) {
        if (command == exitCommand) {
            DiscoveryManager dm = DiscoveryManager.getInstance();
            dm.removeTargetListener(this, TargetType.NDEF_TAG);
            destroyApp(false);
            notifyDestroyed();
        }

    }

                                                       向用户显示提示消息
    private void displayAlert(String error, AlertType type) {  <-
        Alert err = new Alert(form.getTitle(), error, null, type);
        Display.getDisplay(this).setCurrent(err, form);
    }
}
```

以上代码展示了如何在一个支持 NFC 的手机上读取外部 NFC 标签的 UID。这段代码的主要类❶首先创建了 GUI 要素❷，然后为这些 GUI 要素附加上命令❸。

当用户激活该命令，代码就会创建 DiscoveryManager❹，开始搜索。如果没有找到目标，代码就会返回❺；如果有就会将目标显示给用户❻。

有趣的是 Google 在 2010 年采用了一个非常相似的方法，发出超过 10 万个 QR 的代码便签(如图 5-16 所示）给美国商户，让消费者 “收藏”某些地方。用户可以用手机扫描 QR 代码，也可以读到特定地点的评论。逐渐地，人们也开始尝试用新的像 Google Goggles（可以在 Android 设备上使用）的图像识别技术，越过 QR 代码扫描程序，获取同样的信息（其结果精度可变）。

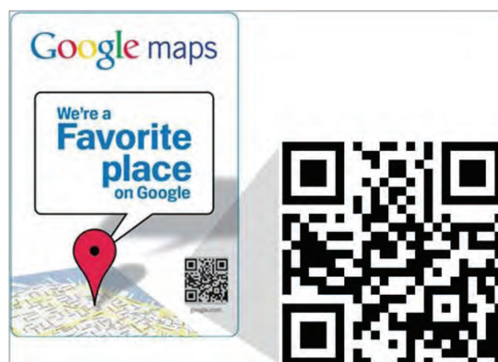


图5-16 Google采用了与Friendticker相似的策略，在2010年发送逾10万个QR便签给美国当地的商户，尝试为虚拟世界和现实世界架起桥梁

类似地，Groupon（2010 年美国成长最快的公司）将其网络策略更进一步，允许用户下载一种移动应用程序（首先是 iPhone，然后 Android 也可以了），根据移动设备（如图 5-17 所示）所在的位置，递送当地的“Groupons”或 Group Coupons 的特价优惠信息。2010 年 8 月该公司为 Gap 推出 Groupon 促销信息，尽管这种活动在波士顿市中心只举行一天，却导致近 50 万的采购者参加了这个 25 美元的特殊折扣活动。

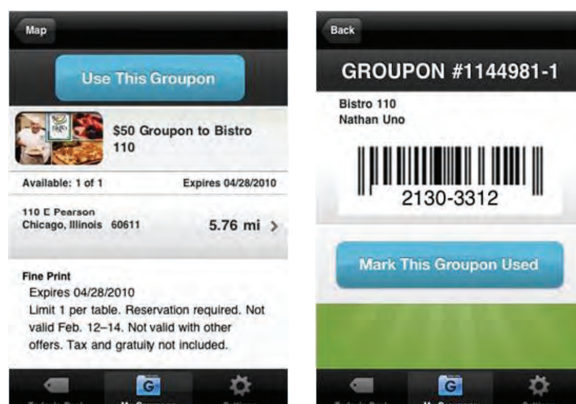


图5-17 Groupon的iPhone应用，用户可以寻找特价优惠信息，或者通过移动设备检测其位置附近的Groupons

现在我们已经了解了实用导航消费者应用和社交应用，下面将会了解 LBS 游戏应用。尽管曾被认为市场空间有限，但很多跨界应用中已经出现越来越多的游戏因素。本章的目的就是要揭示前面讨论过的所谓长尾市场。此外，基于位置的社交网络游戏的成功，证明新市场能快速转变为大众市场。接下来，我们看看这个爆发的领域。

5.3　娱乐或游戏

新一代基于位置的游戏可以分为两种[①]：

- 位置感知游戏
- 空间感知游戏

位置感知游戏包含了游戏中一个玩家的位置信息。一个典型的例子可能就是玩家必须到达一个特定地点，在那里寻找财富。

空间感知游戏将现实中的环境引入游戏当中。这就为现实世界和虚拟世界创建了一个联系。基于位置的移动游戏 Mobile Hunters 就属于这种类型。

接下来我们将分别介绍位置感知游戏和空间感知游戏的例子。这两种游戏的共同点就是它们都要依赖于移动设备上的 GPS，比较费电。除了 iPhone 手机上的游戏，这两种游戏都需要备用的电池组或单独的 GPS 设备。

第三方社交网络整合

对于现存的基于位置的移动社交网络，尤其是新近推出的产品而言，将公共 API 与流行的第三方社交网络整合势在必行。Facebook 拥有 5 亿的全球用户，而 Twitter 有超过 8 千万的全球用户，这就意味着在开发一种新应用时，英语产品不可能无视它们的存在。从前面的例子可以看出，不与广泛普及的社交网络整合的服务非常少，不参与整合就要冒很大风险。例如 Facebook Connect,开发人员可以利用一系列的 API 将第三方服务与 Facebook 相整合，这样就能为应用增加丰富的社会情境。

5.3.1　位置感知冒险游戏GPS Mission

GPS Mission 是一个位置感知游戏，能在 JME 手机和 iPhone 上使用，它能检测用户的位置并据以给出任务（如图 5-18 所示）。在附近没有任务时用户可以通过直接访问网站创建任务。

GPS Mission 在 Map 标签中向用户显示他们周围的世界，列出激活的任务（也许是在特定地点猜个谜语）或该区域内（没有激活任务的情况下）可以收集的“黄金”。

这种游戏最主要的缺点是不能在移动设备上创建任务，因此限制了在户外使用时的便携性。

① Nicklas, D. et al.，“Towards Location-based Games,”21 世纪电脑游戏应用与开发国际会议论文集：ADCOG21（中国香港行政特区，2001），61-67。

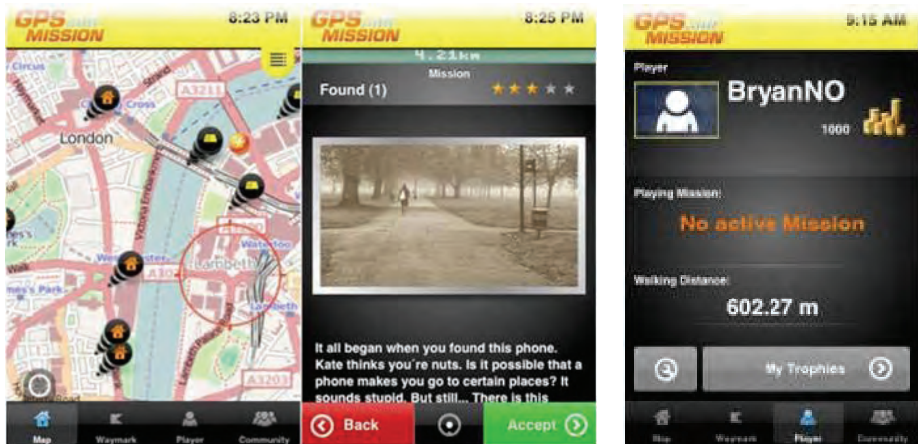


图5-18　GPS Mission iPhone应用，根据游戏者位置给出任务

这种游戏有很多有意思的地方，其中一个特色是 Activity Stream，游戏者可以在任何时段看到世界上哪些地方正在执行什么样的 GPS 任务，比如在巴西玩音乐恶作剧，在中国台湾玩运动大挑战，或者在澳大利亚有照片旅行。另外，你也可以选择与其他人实时分享游戏经验，不仅可与GPS团体中的成员分享，而且还可以与Facebook上所有的朋友分享。最后，你可以利用Waymark功能，在任何地方留下文本和照片信息。

5.3.2　位置感知寻宝藏游戏Geocaching

Geocaching 是一种搜寻宝藏的高科技空间感知游戏，装备了 GPS 设备的冒险者可以在世界任何地方参加这种游戏。该游戏的基础概念就是放置好需要隐藏的容器，叫做 geocaches，在户外用 GPS 查明该容器的准确位置，然后让其他寻宝者来搜寻[①]。Geocaches 是由 Geocaching 创建的，该游戏尝试在基于位置的游戏中融入强烈的团体感和环境支持。

这种游戏按困难程度的不同划分为不同的级别，与 GPS Mission 典型的城市环境相反，它倾向于将游戏地点设置在开放的野外环境之中。另一个不同的地方就是 geocaches 中包含了真实存在的物体。游戏者一旦找到目标，还有责任把它放回原地，这样游戏者在寻宝日志上就能有第一个进入的记录。

在 Geocache 中能够放置一些特定的物体，包括 Geocoins 和 Travel Bug，当游戏参与者将其移走时，就要进行追踪。总的来说，这个游戏要求参与者具备大量的技巧，通常会要求使用者在充满挑战的地方（包括在水下）搜寻，这些特点为此游戏赢得了很多用户。

图 5-19 展示了能在 iPhone 上花费 9.99 美元购买的一个 Geocaching 应用，该应用的主要特点如下：

① Jörg Lonthoff 和 Erich Ortner，“Mobile Location-Based Gaming as Driver for Location-Based Services (LBS) Exemplified by Mobile Hunters，”达姆施塔特大学，2007 年 3 月。



图5-19　Geocaching iPhone应用截图，游戏者可以在他们的邻近区域内操作来找到隐藏的geocaches

- 通过当前的位置、地址或 geocache 代码搜寻
- 访问 geocache 详细资料，包括描述、最近的日志、提示和详细目录
- 寻找能跟踪到的物品，包括跟踪时的目标
- 记录 geocache 发现成果，并在此区域内粘贴注释
- 下载 Pocket Queries，以便离线时使用

介绍过两个游戏之后，接下来再看看这些游戏的最新进展，主要是整合社交网络元素的情况。实际上，这些新服务的成功是非常引人注目的，已经创造了能与 Facebook 这样的社交网络相抗衡的规模。

5.3.3　基于位置的社交网络游戏

鉴于移动社交网络领域中的游戏越来越多，游戏公司正在寻求一种方法，通过引进移动社交网络应用标准的游戏元素实现游戏的多样化。其额外的好处就是这种游戏元素使得团体变得更紧密，因为它鼓励参与者重复冲击排行榜或者提高特定成员的排名。

此领域内的公司数量迅速增长，正在超过单纯的基于位置的游戏。

三种比较受欢迎的服务是 Foursquare、Gowalla 和 MyTown。

移动社交网络游戏 FOURSQUARE

Foursquare 是在 2009 年推出的一款游戏，旨在将游戏元素引进基于位置的社交网络。该游戏主要是为了在竞争越来越激烈的领域内，吸引更多用户的注意力。到 2011 年，其用户数量估计

可达四百万。Foursquare 应用能在 Android、iPhone 和 Blackberry 设备上运行。

Foursquare 的设计思想是奖励参加城市探索的用户。怎么奖励？用户每一次签到，或者每一次登录，都会获得一点数。这有赖于用户是否进入新地方或者鲜有人发现的地方，这样用户就能获得一系列不同的虚拟勋章，并表现在游戏中参与者的外观上（如图 5-20 所示）。如果用户是某个特定地点签到次数最多的人，那么他们就会成为该地点的虚拟“市长”（除非有其他人能够超过其签到次数）。

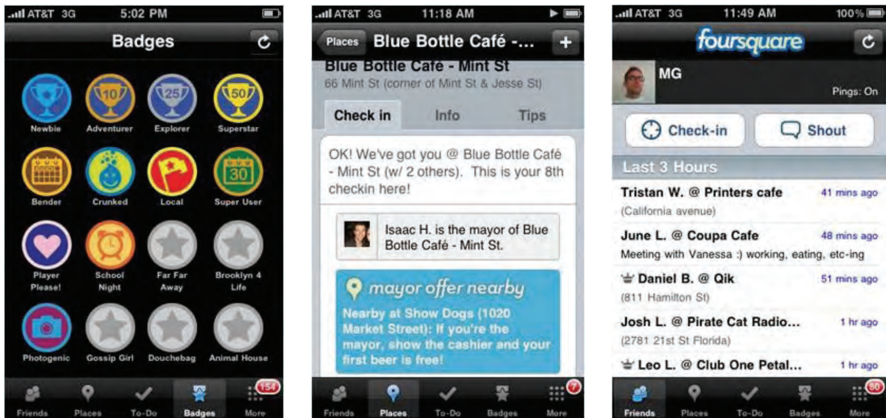


图5-20　Foursquare截图展示了用户签到按钮、在本地可用的特殊服务，以及通过挣得点数就可以获得的勋章

Foursquare 点数是根据地点是不是新地点发放的，而专门的 Discoverer Bonus 点数就是用来奖励用户在一个新地点的签到行为。额外的积分也会奖励一天中多次在相同地点签到或者连续多次签到的行为。

Foursquare 的 iPhone 应用最近做了更新，可以向某些徽章的所有者或者获得“市长”头衔的用户给出一些特别优惠的提示，从而将现实和虚拟世界联系起来。游戏参与者也可以实时发现当地的促销信息，例如某些商品的折扣代码，以及因为定期出入相同的地方而获得忠诚奖励。

虚拟的大富翁游戏 MYTOWN

MyTown 的 iPhone 应用（估计在 2010 年年初有 50 万用户）是一种基于位置的游戏，这种游戏成功地将自己与 Foursquare 区分开来。Foursquare 的签到更侧重于社交方面的功能，MyTown 更多的是一款真正的游戏。事实上，它被人说成是虚拟现实中的“大富翁”游戏。

虽然这种游戏的核心思想还是要求用户登录到某个地方，但驱使玩家这样做的动力却是能够“购买”并且开发道具，而不是告诉朋友你在哪里。例如，如果你登录到一个咖啡馆，就可以买下它，那么其他登录到这的人就会付给你租金。此游戏的目标就是积累尽可能多的道具和财富。你也可以观察人们都聚集到哪个场所，以及根据登录次数评定的该场所的流行等级。

MyTown 2.0（如图 5-21 所示）还创建了一个虚拟商店。在这里，用户可以花费在游戏中收集的虚拟货币来购买各种收藏品。其他 MyTown 正在开发的产品与 Foursquare 和 Gowalla（参看

下一节）非常接近，就是现实世界的促销信息。当你登录到一个场所或在其附近时，如果附近有特价商品交易，这个游戏就会给你发送提示信息。例如，MyTown 已经与 Quiznos 签了协议，支持某些地点轮流降价。

知道 MyTown 后，我们来了解一下 Gowalla 稍许不同的方法，它利用众包来获取内容。



图5-21　MyTown iPhone应用截图展示了Quiznos的位置及其位置统计以及用户登录此场所可以获取的奖品

程序内购买

MyTown 有一个程序内购买功能，用户可以在游戏中购买“攻击力”，这样就能使游戏玩起来更容易。例如，一个“攻击力”可以让用户从自己的建筑物中自动收集租金。这是此游戏的关键部分，没有这种“攻击力”，用户就只能耗费时间手动收集了。你也可以购买更多的财产所有权以及其他的东西，从而扩张统治地盘。目前这些程序内购买的价位在 0.99 美元到 9.99 美元。

融合众包的游戏 GOWALLA

Gowalla 提供的 iPhone 应用与 Foursquare 相似，用户可以在到达某个地点后使用签到功能，然后通知友人，比如到了一个饭店或酒吧可以挣得“徽章”。最近，Gowalla（如图 5-22 所示）引进了一种功能，点击一个场所就可以看到在此地排名前十位的人员名单。

Gowalla 与其同行的关键不同，就是其基于位置的信息是完全众包的。这就使应用程序商店能快速地在世界各地吸引用户，然后快速地建立位置数据（每天能创建超过 1 千个的新位置）。

与 MyTown 一致，Gowalla 也计划支持虚拟商品的程序内购买，除了与大品牌的广告客户签约之外，其赢利计划中大量的要素都有赖于这些虚拟商品。

我们已经知道了每个消费者需求领域的关键实例：导航需求、联系他人的需求，以及休闲娱乐需求。这使我们了解到多数基于位置的应用主要可以分为三类。在此基础之上，我们就可以知

道消费者需求未来的前景，侧重于利用手机内置的罗盘和视频功能进行用户互动。下一节讨论实现这种互动需求的增强现实应用（以及功能）。

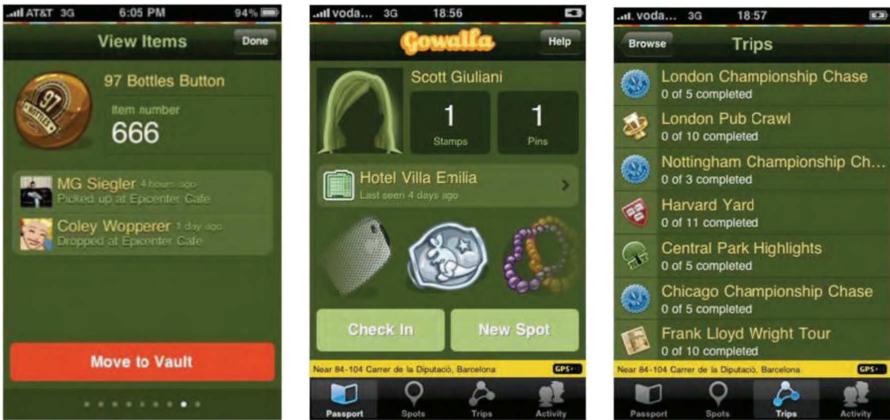


图5-22　显示用户情况、某位置可用物品及特定地方游玩项目的Gowalla截图

5.4　前沿应用

在本节中，我们将会了解智能手机的普及为位置感知应用开发人员带来的机遇。特别介绍新手机上可能支持的增强现实（AR）应用，并了解 Acrossair，尤其是 Layar 的 AR 应用实例。Layar（由 Intel Capital 创建）创建了 AR 应用的标准，推出了很多符合此标准的有趣功能。尽管现在 Layar 仍是最受关注的服务，但是 Junaio 和 Metaio 以及其他很多 AR 公司也加入了竞争。在了解 Layar 实例前，让我们先看看什么是增强现实。

5.4.1　增强现实

从 2009 年中旬开始，随着新型手机具备（特别是 Android 设备）监测方向和倾斜度的能力之后，AR 应用就引起热议，因为一种全新的应用模式诞生了。

增强现实是指将现实中动态或静态的景观与数字影像和信息相结合，比较典型的就是叠加层（如图 5-23 中 Wikitude 应用的例子）。早期著名的 AR 实例来自电影产业。在 2003 年的电影《终结者 3》中，由 Arnold Schwarzenegger 饰演的角色一丝不挂地来到地球，利用 AR 能力扫描人类，以寻求合身的衣服。

尽管移动 AR 领域正处于移动开发的初级阶段，自从 Wikitude 推出 AR 平台以及 Android 推出 World Browser 之后，社会上就涌现了很多相关的应用产品及初创企业。



图5-23　Wikitude是AR领域早期的开拓者，它在2009年中旬为Android设备推出了AR浏览器（展示真实的情节）。此浏览器将Wikitude和其他用户产生的内容放在手机的照相机视图中（来源：mng.bz/qKp5）

5.4.2　Layar：AR 浏览的拓荒者

Layar 在 2009 年 6 月发布了适合 Android 的 AR 浏览器（如图 5-24 和图 5-25 所示），然后在 2010 年推出了适合 iPhone 的 AR 浏览器，接着就是 Acrossair 的 AR 浏览器（如图 5-26 所示），而且也有 iPhone 版。日本的开发商 Tonchidot 创造了一种叫作 Sekai Camera 的应用，将 AR 功能与某些社交网络的功能结合起来。



图5-24　Android上的Layar应用截图，用户可以将视频图片与有用信息结合起来，这些信息作为一个数字图层显示在屏幕视图上



图5-25　Android上的Layar应用的第二张截图，在纽约市中心的视频图像上显示着Wikipedia信息

使用 Layar，用户手机就像是一个能够实时获取显示信息的相机，将内容图层显示在视频图像上面。Layar 软件从手机的 GPS、加速计以及指南针上获取信息，从而检测用户的位置及其方向，这样就能将返回的信息显示在气泡图中，包括到特定地标或者兴趣点之间的距离。

当前的 AR 应用，比如 Layar，还不能识别从手机照相机上获取的影像资料，倒是能利用定位技术来“理解”用户的位置以及用户通常看到的内容。也能生成一个 POI 来“作弊”，得到与用户位置无关的其他位置的信息。相应 POI 实例的服务器代码[①]如下所示。

代码清单 5-2　为一个 Layar AR Android 应用图层服务的 POI 实例

```
<?php

$lat = $_GET['lat'];

$lon = $_GET['lon'];

define( 'LAYERNAME', '' );
define( 'IMAGEURL', '' );
define( 'BASEURL', '' );
define( 'MODEL', '' );

$hotspot = array(
    'title'        => 'My first POI',
    'line2'        => 'Layar Reality Browser',
    'line3'        => 'example application',
    'line4'        => '',
    'attribution'  => 'www.layar.com',
    'type'        => 0,
    'lat'          => ($lat + 0.0005) * 1000000.0,
    'lon'          => $lon * 1000000.0,
    'dimension'    => 1,
```

❶ 取得经、纬度

❷ 定义一些值

❸ 创建 POI 数组

❹ 在经、纬度标识的地点放置 POI

① 2010 年由当时 Layar 的质量保证经理 Marc Rene Gardeya 友情提供。

```
        'object'      => array(
            'baseURL' => BASEURL,
            'full'    => MODEL,
            'reduced' => '',
            'icon'    => '',
            'size'    => 20
        ),

        'transform'    => array(
            'angle'  => 0,
            'rel'    => FALSE,
            'scale'  => 10
        ),

        'actions'      => array(
            array(
                'label' => 'Go to website',
                'uri'  => 'http://www.layar.com'
            )
        ),

        'distance'    => 0,
        'imageURL'    => IMAGEURL,
        'id'          => 'POI_1'
    );

    $hotspots = array( $hotspot );

    $response = array(                     ❺ 创建响应数组
        'nextPageKey' => '',
        'morePages'  => FALSE,
        'hotspots'    => $hotspots,
        'layer'      => LAYERNAME,
        'errorCode'  => 0,
        'errorString' => ''
    );
      $json = json_encode($response);
       header( 'Content-type:application/json');   ❻ 将数组编码为 JSON
       echo $json;
    ?>
```

这段代码为 Layar 应用提供了一个 POI 示例图层。PHP 代码首先从引入的请求中获得经纬度❶，然后设定一些常量值❷。接下来创建了一个数组❸，保存放在经/纬度的 POI 值❹。然后用 JSON 编码❺将这些值作为一个 JSON 串❻送回。

在图 5-25 中，可以看到纽约市中心增强现实的截图，此图上的兴趣点是从 Wikipedia 获取的。

Layar（还有它的竞争者）提供一种 API，能够让开发人员利用 Layar 浏览器创建自己的 spin-off 应用。代码清单 5-3 展示了一个完整的 POI 请求/响应示例，它作为 Layar 图层的一个部分被展示出来。

代码清单 5-3 利用 Layar Android AR 浏览器 API 的 API 响应

```
Request (no OAuth signing)                    ❶ 向服务器请求数据

The developer ID is 896, the developer key wpvm1f4g. Appending the timestamp
     1249226148713 and hashing with SHA-1 will generate the hash shown in the
     request.

http://devAPI.example.com/getPOIs/?countryCode=IN
&lon=4.887339&timestamp=1249226148713
&userId=ed48067cda8e1b985dbb8ff3653a2da4fd490a37
&developerId=896&developerHash=1ee6d294aa6b639b365899f844257523c5bf9702
&RADIOLIST=a&radius=6245&CUSTOM_SLIDER=23987.0
&lat=52.377544&layerName=snowy4
&SEARCHBOX=asdfdhcgg&accuracy=100              ❷ 服务器返回数据

Response
```


```
{"hotspots": [{"distance": 100, "attribution": "This is a test layer POI
     provider", "title": "My layer - snowy4, location = 52.377544, 4.887339",
     "lon": 4884339, "imageURL": null, "line4": "RADIOLIST-None,CustSlider-
None", "line3": "SEARCHBOX - asdfdgxdg", "line2": "DevlId - 896Settings:
range=1000", "actions": [], "lat": 52374544, "type": 0, "id": "test_1"},
{"distance": 100, "attribution": "This is a test layer POI provider",
"title": "My layer - snowy4, location = 52.377544, 4.887339", "lon":
4887339, "imageURL": null, "line4": "RADIOLIST-None,CustSlider-None",
"line3": "SEARCHBOX - asdfdgxdg", "line2": "DevlId - 896Settings:
range=1000", "actions": [], "lat": 52374544, "type": 0, "id": "test_2"},
{"distance": 100, "attribution": "This is a test layer POI provider",
"title": "My layer - snowy4, location = 52.377544, 4.887339", "lon":
4890339, "imageURL": null, "line4": "RADIOLIST-None,CustSlider-None",
"line3": "SEARCHBOX - asdfdgxdg", "line2": "DevlId - 896Settings:
range=1000", "actions": [], "lat": 52374544, "type": 0, "id": "test_3"},
{"distance": 100, "attribution": "This is a test layer POI provider",
"title": "My layer - snowy4, location = 52.377544, 4.887339", "lon":
4884339, "imageURL": null, "line4": "RADIOLIST-None,CustSlider-None",
"line3": "SEARCHBOX - asdfdgxdg", "line2": "DevlId - 896Settings:
range=1000", "actions": [], "lat": 52377544, "type": 0, "id": "test_4"},
{"distance": 100, "attribution": "This is a test layer POI provider",
"title": "My layer - snowy4, location = 52.377544, 4.887339", "lon":
4887339, "imageURL": null, "line4": "RADIOLIST-None,CustSlider-None",
"line3": "SEARCHBOX - asdfdgxdg", "line2": "DevlId - 896Settings:
range=1000", "actions": [], "lat": 52377544, "type": 0, "id": "test_5"},
{"distance": 100, "attribution": "This is a test layer POI provider",
"title": "My layer - snowy4, location = 52.377544, 4.887339", "lon":
4890339, "imageURL": null, "line4": "RADIOLIST-None,CustSlider-None",
"line3": "SEARCHBOX - asdfdgxdg", "line2": "DevlId - 896Settings:
range=1000", "actions": [], "lat": 52377544, "type": 0, "id": "test_6"},
{"distance": 100, "attribution": "This is a test layer POI provider",
"title": "My layer - snowy4, location = 52.377544, 4.887339", "lon":
4884339, "imageURL": null, "line4": "RADIOLIST-None,CustSlider-None",
"line3": "SEARCHBOX - asdfdgxdg", "line2": "DevlId - 896Settings:
range=1000", "actions": [], "lat": 52380544, "type": 0, "id": "test_7"},
{"distance": 100, "attribution": "This is a test layer POI provider",
```


5

```
"title": "My layer - snowy4, location = 52.377544, 4.887339", "lon":
4887339, "imageURL": null, "line4": "RADIOLIST-None,CustSlider-None",
"line3": "SEARCHBOX - asdfdgxdg", "line2": "DevlId - 896Settings:
range=1000", "actions": [], "lat": 52380544, "type": 0, "id": "test_8"},
{"distance": 100, "attribution": "This is a test layer POI provider",
"title": "My layer - snowy4, location = 52.377544, 4.887339", "lon":
4890339, "imageURL": null, "line4": "RADIOLIST-None,CustSlider-None",
"line3": "SEARCHBOX - asdfdgxdg", "line2": "DevlId - 896Settings:
range=1000", "actions": [], "lat": 52380544, "type": 0, "id":
"test_9"}], "layer": "snowy4", "errorString": "ok", "morePages": false,
"errorCode": 0, "nextPageKey": null}
```


这段代码展示了如何调用 Layar API 取得给定区域的兴趣点。一旦调用❶，代码就会将特定的经纬度及其他参数传递给服务器。然后服务器就会返回特定点附近所有的 POI❷，这些 POI 将被显示在地图上。

Acrossair 开发了自己特有的 AR 新产品，侧重于寻找地铁站，用于帮助地面上的用户找到正确的地铁站。可以参看图 5-26 所示的例子。



图5-26 Acrossair的地铁寻找AR应用，2009年夏天发布iPhone 3GS版，是早期针对iPhone平台的AR应用程序之一

Layar API 和 APK（Android Package File）的完整描述及更多代码例子可以在 http://layar.pbworks.com 上找到。

最后一节，我们快速地展望了一下未来，以便你能感受位置感知应用的种种可能。Layar 推出的哪类服务（以及呈现形式）对未来开发新的、令人兴奋的产品有着重要的启发作用。最近的一项进展就是将摄像头获取的影像首先用于识别，然后与二维物体互动。这种技术可用于增强印刷广告效果，能触发屏幕上的三维动画影像或者短视频。

当用户在某点走来走去时，哪些 POI 是用户可见的？

对于像 Layar 这样的 Android 应用，移动客户端将只显示返回的 POI，并在用户范围滑块划定的特定范围之内（或者要是没有滑块，就在用户周围 1500m 的范围之内）显示。利用 OS 客户端提供的 API 来计算用户与 POI 之间的距离。目前，Android 上使用的是 WGS 84 椭球，这可能会导致其计算结果与 Haversine Formula（利用完美的球体计算）的计算结果稍有不同。但是由于通常使用的范围都非常小，因此两者的不同是可以忽略的。

5.5 小结

手机上的位置功能越来越多，不仅能为消费者提供大量有意思的基于位置的应用，还意味着手机将替代辅助导航以及 PND 公司。本章介绍了不同领域的消费者应用，从导航到一个地方的应用到休闲娱乐应用，但实际上不同应用的功能相互整合更能为消费者提供价值。现在移动导航应用包括了社交网络元素，反之亦然。类似地，社交网络也包含了游戏元素以便为社区用户创造交往的机会。

随着新型手机具有了先进的加速度计和指南针功能，手机大体上是向着更智能的方向发展，不过此领域内的功能探索和创新仍有着巨大的空间。增强现实应用开发者们对这个新领域的开发才刚刚开始，所有的元素开始汇集于此，以期开发出革命性的智能应用。

辅助导航公司通过技术革新、迎合市场需求，首先成功地建立了盈利模式。在 Google 这样的主流企业的推动下，原有的市场情势早已被改变。这对开发商和企业主来说甚至更为重要，他们不仅要考虑怎样才能把应用转化为财富，而且还要革新收费的方法和对象。我们将会在第 8 章详细介绍这个方面。

从本章中可以看到，有大量的功能可以嵌入到基于位置的应用之中。或许我们所做的还仅仅是揭示其冰山一角。然而，移动开发的复杂性意味着不是所有的应用功能或服务都能以相同的方式用在移动平台上。因此掌握每一个移动平台的适用范围（也是局限性）是非常基本的要求。

下一章中，我们会了解对开发人员来说，依据他们计划构建的位置功能或位置感知应用程序类型，选择哪种移动平台更切实有效。

第 6 章 移动平台

本章内容

- 最新移动开发趋势
- 主要移动开发平台比较
- 移动开发框架
- 移动应用测试

苹果公司与谷歌公司为我们带来了 iPhone、iPad 及 Android 这样的设备及系统，从而使移动开发领域产生了深刻的变革，从一个封闭的 API、以电信运营商为中心的模式转变为开放的 API、以设备制造商和软件开发商为中心的模式。对 LBS 应用程序开发商而言，这种转变也代表着巨大的商机，因为现在可以开发功能齐全的 LBS 应用，并直接将其发布给用户——至少多数时候是这样的。

但是，移动开发只是在若干不同移动运营平台之间并不完整的开发，每一种平台所支持的功能和应用范围都各不相同。因为可选择的移动平台有很多，所以在规划设计 LBS 应用时就必须要了解每个移动平台的市场趋势。另外，由于移动电话上的资源有着诸多限制，（这包括屏幕大小、内存、CPU、存储容量以及输入方法等方面），从而使得正确理解并选择合适的开发技术变得尤为重要。图 6-1 展示了主要的智能手机平台。

在本章中，我们要了解一下从 iPhone 和 Android 到相对较新的 MeeGo 这些主要的移动开发平台，并介绍每个平台的基本开发过程，以便大致了解每个平台的利弊。

但是在展开分析之前，为了解当前的市场动态，我们先来介绍一下市场是如何演变的以及哪种平台比较受欢迎。

6.1 移动电话行业动态

据高德纳公司统计，2010 年第三季度移动电话销售量为 4 亿 1700 万部，比 2009 年同期增加了 35%[1]。诺基亚公司继续占有市场领导地位，但是其所占份额从 2009 年第三季度的 36.7%降到

① http://www.gartner.com/it/page.jsp?id=985912

了 2010 年同期的 28.2%（参见表 6-1）。三星公司仍排在第二位，其市场份额比去年同期有所增加，销售数额总计为 7160 万部。



图 6-1 瓦解旧有移动生态系统的几大主要移动平台：iPhone、 Android、HP webOS、BlackBerry Storm 以及 iPad

表6-1 2010年第三季度全球移动电话销售情况

公　　司	2010年第三季度销售量（千部）	2010年第三季度份额%
Nokia	117 461.0	28.2
三星	71 671.8	17.2
LG	27 478.7	6.6
苹果	13 484.4	3.2
RIM	11 908.3	2.9
索尼爱立信	10 346.5	2.5
摩托罗拉	8 961.4	2.1
HTC	6 494.3	1.6
ZTE	6 003.6	1.4
华为科技	5 478.1	1.3
其他品牌	137 797.6	33
合计	417 085.7	100

智能手机去年第三季度的销售额增长了 96%，占 2010 年第三季度全部手机销售量的 19.3%（参见表 6-2）。

表6-2　2010年第三季度按操作系统统计的全球智能手机销售情况

公　　司	2010年第三季度销售量（千部）	2010年第三季度份额（%）
Symbian	29 480	36.6
Android	20 500	25.5
iOS	13 484	16.7
Research In Motion	11 908	14.8
Microsoft Windows Mobile	2 247	2.8
Linux	1 697	2.1
其他	1214.8	1.5
合计	80 532	100

在以上介绍的这种发展背景下，移动开发技术是如何演进的呢？下文就将介绍这一点。

6.2　移动开发技术与发行方式的变化

iPhone 出现以前，运营商决定着一切。产品开发需要很长的时间，产品部署则需要更长的时间。苹果公司现已成为旧有移动生态系统的主要颠覆者，运营商颐指气使的黄金年代一去不复返了。曾经高昂的移动开发准入费用已几乎完全取消了。在 Google 推出 Android 及其他打破该垄断格局力量不断涌入的情况下，任何人都可以利用开放的 API 及发行平台上（应用程序商店）进行移动开发了。

虽然 Java 仍是很多移动应用使用最多的一种格式（尤其在移动营销和移动游戏这类应用中），但像 iPhone 这样更精巧的平台正愈加受到人们的推崇。而前所未有的是，Android 还特别允诺会免费公布其移动操作系统，允许用户完全整合手机功能及在其上运行的应用。

另外，将来会出现更多的应用程序商店，从而显著改变移动应用开发业务的现状。开发者们不再需要运营商的首肯，这种巨大的改变重塑了移动行业。开发者们可以直接面向用户开发并发布其应用。（在表 6-3 中展示了所有现存的应用程序商店概况，在第 11 章中会详细介绍各个应用程序商店。）

表6-3　移动应用程序商店（第11章会更详细地介绍）

名　　称	成立日期（年/月/日）	应用程序数量（款）	下载量（亿）	开发平台	开发者份额（%）	开发应用所需费用
GetJar	2004	53 000	6	多平台	N/A	免费
Apple App Store	2008/7/10	75 000	20	iOS	70	99美元
Android Market	2008/10/22	10 200	未知	Android	70	25美元

（续）

名　　称	成立日期（年/月/日）	应用程序数量（款）	下载量（亿）	开发平台	开发者份额（%）	开发应用所需费用
Software Store	2008/12/16	5 000	未知	Palm OS Windows Mobile	60	未知
BlackBerry App World	2009/4/1	2 000	未知	BlackBerry OS	80	200美元
Ovi Store	2009/5/26	未知	未知	Symbian	70	50欧元
App Catalog	2009/6/6	41	160 000	WebOS	70	免费
LG Application Store	2009/7/13	1 400	未知	Windows Mobile	N/A	N/A
Samsung Application Store	2009/9/14	未知	未知	Java Symbian Windows Mobile	70	1美元
Windows Marketplace for Mobile	2009/10/6	N/A	N/A	Windows Mobile	70	99美元/年

在回顾了现有移动开发及发布平台的发展情况后，下面我们将介绍一下现今所有适用的开发平台。

6.3 Java ME

Java Platform, Micro Edition（Java ME——以前被称作 Java 2Platform，Micro Edition，或者简称为 J2ME）是专为移动设备和内嵌系统设计的一种 Java 平台，适用于移动电话、导航设备，以及机顶盒设备。图 6-2 展示了 Java ME 的架构。

Java ME 是一种可移植的解决方案，能为多种设备提供通用的函数库，其开发的应用程序能在多种不同的设备上运行。但是多数手机上运行的应用程序（包括其数据）其大小一般不会超过 1 MB。另外，为了有效使用一些特定的 API，比如文件系统准入 API，它们还必须进行加密处理。为了便利加密签名过程，行业内还成立了 Java Verified 机构（Java 移动代码签名证书认证机构）。

提示　JAVA VERIFIED　Java Verified[①] 是一个全球性的组织机构，其遍及全球的移动系统成员通过分工协作来处理 Java ME 的。该机构能向 Java ME 开发人员提供测试功能及开放的资源，用于帮助他们更快地为更多设备开发高质量的 Java ME 应用程序。Java Verified 管理委员会由来自 LG、摩托罗拉、诺基亚、Orange、三星、索尼爱立信、Sun Microsystems 及 Vodafone 的代表组成。

① http://javaverified.com

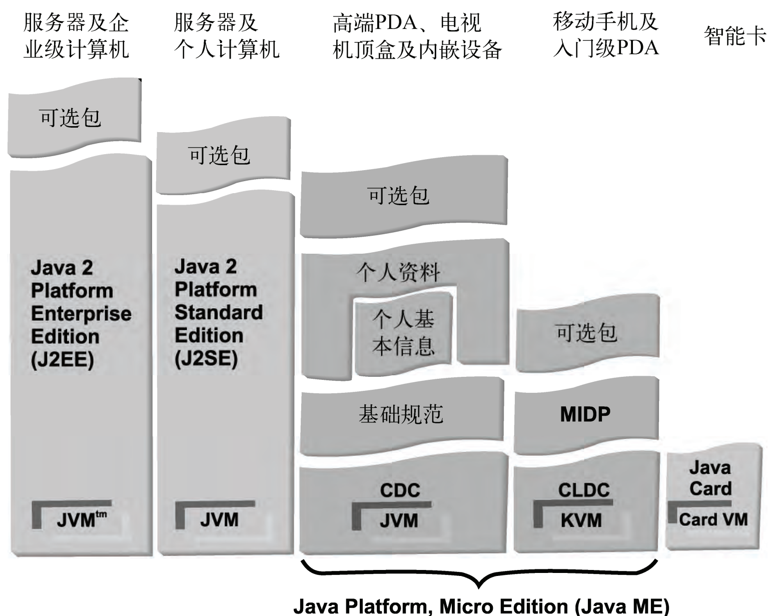


图6-2　Java ME平台是整个Java平台的一个子集，其面向目标是移动设备（来源：mng.bz/8aLR）

Java ME 应用程序能在 BlackBerry、Symbian 和多数支持 Java 语言的手机上运行。它便于应用走向大众。对不要求特定平台原生程序库的高端应用程序开发来说，Java ME 是一种很好的选择。如果你设计的 Java ME 应用程序要访问原生移动平台的 API，比如利用黑莓中的图形硬件显示响应更敏感的用户界面，那么该程序自然就不能在 Symbian 设备上运行。所以假如应用程序采用高端通用 Java ME 语言，那就必须确保该应用能成功运行。如果能成功使用 Java ME，那么这种语言将是很棒的一种解决方案。否则，就不得不稍作调整，在每个 Java 目标平台上编写不同的应用程序。

6.4　Symbian

Symbian 平台是一种为移动设备开发的开源操作系统，在 Nokia、NTT DoCoMo、Sony Ericsson 的软件资产，包括 Symbian OS、S60、 UIQ 和 MOAP(S)用户界面基础上整合而成。图 6-3 展示了基于 Symbian 手机的系统架构。

Symbian 是 Symbian OS 的后续产品，是在 2009 年 4 月 Symbian Foundation 成立后推出的。它能在 S60(第三版或更新的版本)和 Sony Ericsson 手机上运行，Samsung、Siemens 以及 Motorola 手机使用 Symbian UIQ 3.0 或更新的版本。（最近 Motorola 开始选用 Android 平台，其将来的手机都不再使用 Symbian 了。）

Symbian 的本机语言是 C++。基于 Symbian OS 的多种平台给应用程序开发人员提供了很多

SDK（软件开发工具包），来为 Symbian OS 设备开发应用。Symbian 对 C++情有独钟。它能支持多种运行时环境，这样开发人员就能在此平台上选用任何编程语言，比如 Python、Java ME、Flash Lite、Ruby、.NET、Web Runtime (WRT) Widgets 和 Standard C/C++。图 6-3 展示了 Nokia S60 平台的架构。

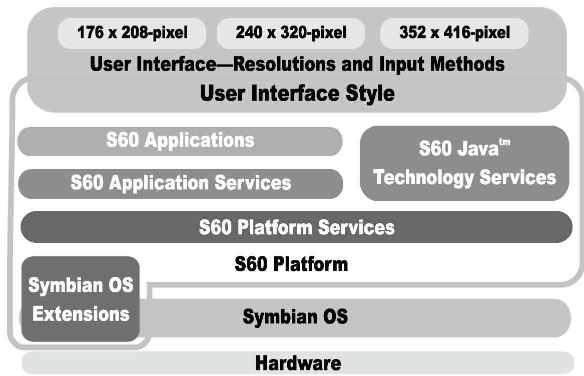


图 6-3 Symbian 是 Nokia S60 平台架构的一个部分（来源：mng.bz/LSKH）

对于平常开发而言，Symbian 是一种功能强大的平台。它从一开始就是为移动设备专门开发的，因此它不仅是一种实时的、多任务开发平台，还为减少内存占用和延长电池寿命专门做了优化。

The Symbian Foundation 负责维护基于 Symbian OS 的开源软件平台，以及由 Nokia、 NTT DoCoMo 和 Sony Ericsson 包括 S60 和 MOAP(S)用户界面提供的软件资产。按 Eclipse Public License 规定，其部分平台源代码已开源。这个程序彻底完成后，其平台代码将会为所有人免费使用。近 3 亿部基于 Symbian OS 的手机将会装载这种代码，占全球市场超过一半的份额。

要开发 Symbian 应用，首先要选择一种目标平台，比如 Sony Ericsson P800、Nokia 3650/7650 或者 Samsung i458， Symbian 只是一种低端的操作系统，一种中间件。每种手机在其基本操作系统上都有不同的 UI（用户界面）。

目前主要有三种 GUI（图形用户界面）系统：

- ❑ Nokia Series 60
- ❑ Nokia Series 80
- ❑ User Interface Quartz (UIQ)

表 6-4 对可用的 GUI 系统做了一个总结。

表6-4 Symbian主要的GUI系统

GUI	特　　性	手机型号
Series 60 v1.x + Symbian OS 6.1	分辨率为 176 × 208 键盘式操作 无触摸屏	Nokia 3650, Nokia 7650, Nokia N-Gage, Siemens SX1, Samsung SGH-D700, Sendo XXX

（续）

GUI	特　　性	手机型号
Series 60 v2.0 + Symbian OS 7.0s	分辨率为176 × 208 键盘式操作 无触摸屏	Nokia 6600
Series 80 + Symbian OS 6.0	分辨率为640 × 200 键盘式操作 无触摸屏	Nokia 9210
UIQ v2.0 + Symbian OS 7.0	分辨率为208 × 320 可选择键盘操作 有触摸屏	Sony Ericsson P800, BenQ P30

Series 60 是最常用的一种平台，最容易上手。选择好目标平台后，需要选择一种开发环境。你可以选择：

- Visual Studio
- Metrowerks CodeWarrior
- Borland C++ Builder
- Carbide.c++（基于 Eclipse IDE，现在属于 Symbian Foundation）

然后可以下载安装特定的来自 Nokia、Sony Ericsson 或其他来源的 SDK（软件开发工具包），之后即可为指定的目标平台开发应用程序了。

Symbian 的用户很多，而且它的开发环境也很完善，但运行在 Symbian 系统上的 GUI 层变体繁多，让人不得不只选择现有 Symbian 用户中的某一部分作为目标。如果你想让自己的应用在大多数 Symbian 手机里运行，恐怕至少要开发 100 个不同的客户端。

由于 Symbian 以 C++作为本机语言并有着复杂的 GUI，对初学者来说，学习过程还是有难度的。另外，从用户角度来说，如果没有针对特定的平台进行优化，用户体验可能会受影响。

对开发人员来说，Symbian 用户繁多也是一个很大的弊端，所以在选择发布平台时，必须权衡可能的开发困难。如果应用程序不需要太多的计算资源，也不用大量处理图形，那么选择 Java ME 也是可以接受的。

有关诺基亚的开源移动 OS: Maemo，请参见 6.12 节。

Nokia Ovi Store

Nokia Ovi Store（如图 6-4 所示）在 2009 年 5 月面向全球推出。从这里用户可以下载移动游戏、应用程序、视频文件、图像和手机铃声。有些项目是免费的；有些可以通过信用卡支付或者通过运营商支付。

Ovi Store 能向用户提供与他们所用的移动设备相匹配的内容，而且与他们的喜好和位置相关。用户可以向他们的朋友推荐这些内容，给朋友看自己下载的内容以及感兴趣的项目。

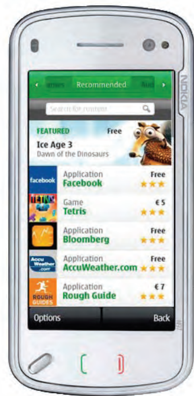


图 6-4 Nokia Ovi Store

对内容发布者而言，Nokia 提供自助服务，用户可以自己将内容发布到 Ovi Store。Ovi Store 能支持的内容包括 J2ME 、Flash 应用、窗体小部件、铃声、墙纸、桌面主题，而 Nokia Series 40 和 S60 手机能支持的内容更多。Nokia 提供 70%的收入分成，但要扣除退款和退货，以及少量可接受的税，还有一点固定的运营费用。更多信息请参考表 6-3 和第 11 章。

6.5 iPhone 和 iPad

Apple的iPhone和iPad永远地改变了移动产业,它们是有史以来卖的最好的消费者电子设备。iPhone 是至今销售最快的一款智能手机，而 iPad 的销售速度甚至超过了 iPhone。它们在开发人员和消费者中的受欢迎程度造成逾 20 万应用产品的迅速问世（2010 年中旬），吸引了全球消费者的注意。下面是 2010 年 5 月 App Store 中的程序数量。

- 活跃程序总量（目前可以下载的）：199 790
- 不活跃程序总量（不能再下载使用的）：34 352
- US App Store 中可见的程序总量：234 142
- US App Store 中活跃的发布者数量：39 283

iPhone 应用程序开发要求使用 Objective-C，苹果开发的 C 的变体，对 C 和 C++的面向对象进行了增强。熟悉 Mac OS X 开发的人员可以快速过渡到 iOS 开发，而由于 Objective-C 和 iOS 的难度，其他开发人员学习起来则要困难得多。

总之，iOS 应用只能在苹果机上用 Xcode 开发，而且要注册为开发人员，用 Objective-C、C 或者 C++和 Cocoa 开发，二进制结果要提交给 Apple App Store 批准，批准时间需要一至两周。让我们看一下 iPhone 和 iPad 应用开发的所有组件：

- iOS——iOS 是苹果公司为 iPhone、iPad 和 iPod touch 开发的操作系统。iOS 应用程序使用 Objective-C 编程语言，利用 Mac OS X 的 Cocoa 应用框架 。它有四个抽象层：Core OS 层、Core Services 层、Media 层和 Cocoa Touch 层（如图 6-5 所示）。

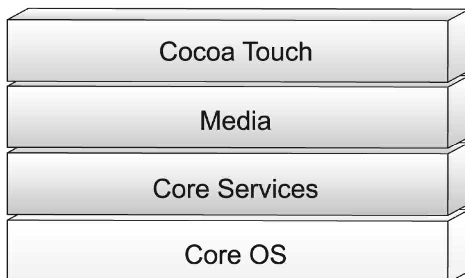


图 6-5　iOS 技术分层示意图（来源：mng.bz/d3j3）

- The iPhone SDK——The iPhone SDK[①]（最初在 2008 年 3 月发布）包含开发和调试 iOS 应用所需的代码、信息和工具。
- Objective-C——Objective-C 是一种反射式的面向对象编程语言，在 C 语言基础上增加了 Smalltalk 式消息传递机制。对初学者来说有些难度。
- Cocoa Touch——Cocoa Touch 是创建能在 iPhone 和 iPod 上运行的软件程序的接口。它提供操作系统的抽象层（如图 6-6 所示）。Cocoa Touch 基于 Cocoa API 工具，用于构建 Mac OS X 软件程序。iPhone SDK 中含有开发 Cocoa Touch 应用程序的工具。

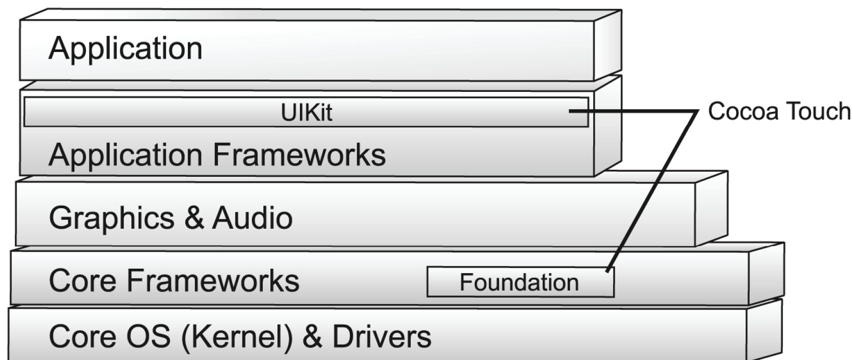


图 6-6　iOS 架构中的 Cocoa Touch（来源：mng.bz/ZGY5）

- Xcode——Xcode 是开发环境。为支持 iOS 开发，Xcode 工具已被更新。除了提供代码的基本编辑、编译和纠错功能外，Xcode 也提供在 iPhone 和 iPod touch 设备上测试用的发射点，而且能在 iPhone 模拟器上运行应用程序；iPhone 模拟器是在本地 Mac 计算机上模拟 iOS 环境的平台。
- The iPhone Developer Program——这个计划能给你提供将代码下载到手机上的密钥。目前个人需花费 99 美元，公司需花费 299 美元。

iPhone 和 iPad 应用开发热潮席卷了全世界，Apple App Store 中就有超过 20 万种的应用程序。

① http://developer.apple.com/iphone/

很多开发人员从其他平台跨界转入 iOS 开发中来，初学者所需的书籍和在线资源也极其丰富。

内存占用少，平台就会比较稳定；耗尽内存的应用很容易崩溃。

iOS4 使 iOS 开发具有了多任务处理能力，但在此之前，使用 iOS 的一个主要限制条件就是有些应用不能在后台运行。推送通知是缓解其中部分问题的一个很好的办法。这样即使用户当时没有运行你的应用，也能给手机传递消息。例如，用户并没有运行聊天应用，但也能收到从朋友那发来的消息。LBS 应用能够向用户传递来自朋友或其他服务的提醒信息。

到目前为止，iPhone 应用程序开发面临的最大问题就是能否通过 App Store 审批，其结果往往是出人意料的，有时还前后矛盾。你开发的应用可能会因为一些不可预知的原因被拒之门外，造成投资血本无归。另一个需要注意的问题在开发人员计划的条款中有说明。Apple 不允许开发人员使用文档中没有列出的功能，为避免此种情况的发生，它会利用自动工具检测来每段代码。我们无法与 Apple 对抗或替代 iPhone 的基本功能。

应用内购买是Apple的另一个重要特色,可以减小用户在你的应用中购买虚拟产品时的摩擦。他们会在 iTunes 账户中看到刚刚购买产品的费用，Apple 会将总收入的 70%直接转账给你。

导致 iPhone 应用开发人员再三抱怨的问题，就是很难在真正的 iPhone 或 iPod Touch 设备上部署应用程序。不管怎样，Apple 为其应用开发人员设计的一些条条框框，引起了一些开发人员的反感和不满，有些人就探索可替代的方法，寻找一种不需要应用认证的非正式的开放工具链。

如果你想让一些评论者免费使用你开发的需要购买的应用产品，可以给他们提供促销码，从而让他们能在 iTunes 上免费下载你的应用程序。

Apple 给开发人员提供了一种最多在 100 部 iPhone 上测试应用程序的方法。如果想在 App Store 接受你的应用之前对应用做些测试，就必须取得这些用户设备的唯一识别号（UDID）。要找到 iPhone 手机的 UDID，插上电脑，等待 iTunes 启动、识别。从 iTunes 的设备列表中选中你的手机，点击 Summary 标签。要看到你的 UDID，点击 iPhone 图片旁边的 serial number。然后，就应该看到字母 identifier 以及一串字符了——这就是你的 UDID。要复制这些字符，在你的电脑上按 Command-Con（或者 Windows 中的 Ctrl-C）。

6.5.1 iOS开发工具的限制条件

Apple 对开发人员创建 iPhone、iPod touch 和 iPad 应用使用的开发工具要求相当严格。恶名昭著的 iPhone Developer Program License Agreement Section 如下：

> 3.3.1——应用开发只能使用文档中提到的 API，并按 Apple 规定的方式，不能使用或调用任何私人 API。应用开发所用的开发语言必须采用 Objective-C、C、C++或 ios Webkit 能执行的 JavaScript，只有用 C、C++和 Objective-C 编写的代码才能编译并与 Documented API 连接（例如，禁止通过中间转换工具、兼容层、兼容工具连接文档中提到的 API）。

这个协议引起了巨大争议，因为这项规定禁止 Adobe 的 Flash 以及其他第三方开发工具成为 iPhone App 转换程序。要想加入 iPhone/iPad 开发，就必须知道 Apple 的所有限制条件和政策。

6.5.2　简单的iPhone应用

现在让我们看看如何创建 iPhone 应用，这个例子用于找到用户的当前位置并显示出来。步骤如下：

(1) 安装最新版本的 iPhone SDK；

(2) Open Xcode（确保有最新版本）；

(3) 转到 File > New Project，创建一个新 View-Based Application 项目，输入项目名称，比如 AroundMyCity。

这样就能为应用创建一个利用单一视图实现用户界面的项目。代码清单 6-1 到代码清单 6-4 展示了这个项目的源代码。第一个代码清单显示 AroundMyCityAppDelegate.h 的源代码，用于实现基于视图的应用 AroundMyCity。

代码清单 6-1　AroundMyCityAppDelegate.h 源代码

```
//
//  AroundMyCityAppDelegate.h
//  AroundMyCity
//
//  Created by Murat Aktihanoglu on 2/19/10.
//

#import <UIKit/UIKit.h>                          // 导入用户界面包头文件

@class AroundMyCityViewController;               // 前向声明类

@interface AroundMyCityAppDelegate
          : NSObject <UIApplicationDelegate> {   // ❶ 从 UIApplicationDelegate 派生类
    UIWindow *window;
    AroundMyCityViewController *viewController;  // ❷ 声明成员变量
}

@property (nonatomic, retain) IBOutlet UIWindow *window;
@property (nonatomic, retain) IBOutlet AroundMyCityViewController
 *viewController;

@end
```

这里我们声明了继承自 UIApplicationDelegate 的一个类❶，为保持 UIWindow 与视图控制器指针一致，还为这个类声明了两个成员❷。

接下来是 AroundMyCity 应用的 AroundMyCityAppDelegate.m 文件的代码。

代码清单 6-2　AroundMyCityAppDelegate.m 的代码

```
//
//  AroundMyCityAppDelegate.m
//  AroundMyCity
//
//  Created by Murat Aktihanoglu on 2/19/10.
//
```

```
#import "AroundMyCityAppDelegate.h"          ← 导入必要的头文件
#import "AroundMyCityViewController.h"

@implementation AroundMyCityAppDelegate

@synthesize window;
@synthesize viewController;
                                                 ❶ 声明重载点以便自定义
- (void)applicationDidFinishLaunching:
    ➥(UIApplication *)application {          ←

    [window addSubview:viewController.view];
    [window makeKeyAndVisible];
}                                                ❷ 声明类的解构方法

- (void)dealloc {                            ←
    [viewController release];
    [window release];
    [super dealloc];
}
```

这段代码实现了 AroundMyCityAppDelegate 类，重载父类的 applicationDidFinishLaunching 函数❶。还提供了一个能够实现用户服务自定义的专门解构方法❷。

下面代码清单包含了 AroundMyCity 应用的 AroundMyCityViewController.h 文件的代码。

代码清单 6-3　AroundMyCityViewController.h 的代码

```
//
//  AroundMyCityViewController.h
//  AroundMyCity
//
//  Created by Murat Aktihanoglu on 2/19/10.
//  Copyright Centrl Inc. 2010. All rights reserved.
//
                                             导入用户界面包头文件
#import <UIKit/UIKit.h>                  ←

@interface AroundMyCityViewController : UIViewController {   ←
                                                              ❶ 声明新类
}

@end
```

这段代码声明了一个源自 UIViewController 的类❶，以便能够实现这个类的自定义版本。

下一个代码清单显示了 AroundMyCity 应用的 AroundMyCityViewController.m 文件的代码。

代码清单 6-4　AroundMyCityViewController.m 的代码

```
//
//  AroundMyCityViewController.m
//  AroundMyCity
//
```

```
//  Created by Murat Aktihanoglu on 2/19/10.
//

#import " AroundMyCityViewController.h"

@implementation AroundMyCityViewController

/*
- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle
    *)nibBundleOrNil {

    if (self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil])
    {
        }    return self;
}
*/

/*
- (void)loadView {
}
*/


/*
- (void)viewDidLoad {
    [super viewDidLoad];
}
*/


/*
- (BOOL)shouldAutorotateToInterfaceOrientation:
       ➥(UIInterfaceOrientation)interfaceOrientation {
    return (interfaceOrientation ==
      UIInterfaceOrientationPortrait);
}
*/

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];

}

- (void)viewDidUnload {
   // e.g. self.myOutlet = nil;
}


- (void)dealloc {
    [super dealloc];
}

@end
```

包含头文件

❶ 初始化方法

❷ 实现 loadView

❸ 实现 viewDidLoad

❹ 重写以支持非默认朝向

返回 Yes 以支持不同朝向

❺ 如果没有超视图则释放视图

释放缓存中未用到的数据

❻ 释放保留的子视图

这段代码利用自定义的❶，❷，❸，❹，❺，❻函数实现 AroundMyCityViewController 类。这些代码多数都是由开发工具创建的，一般在所有项目中都相同。完成这一步后，我们会自定义各个函数，提供所需要的功能，操作如下：

(1) Select Project > Set Active SDK > iPhone Simulator 3.0；

(2) 在应用中添加 Core Location framework，右键点击 Frameworks，选择 Add > Existing Frameworks。选择 CoreLocation.framework；

(3) 选择 Import < CoreLocation/CoreLocation.h > in AroundMyCityController.h。

(4) 对 AroundMyCityController 类进行下列修改，获取用户位置。

代码清单 6-5 和代码清单 6-6 是实现该项目全部功能的最终源代码。

代码清单 6-5　获取当前位置的最终 AroundMyCityController.h 文件代码

```
//
//  AroundMyCityViewController.h
//  AroundMyCity
//
//  Created by Murat Aktihanoglu on 2/19/10.
//                                                    ❶ 包含核心位置头文件

#import <CoreLocation/CoreLocation.h>
#import <CoreLocation/CLLocationManagerDelegate.h>

#import <UIKit/UIKit.h>

@interface AroundMyCityViewController :
     UIViewController<CLLocationManagerDelegate> {

        CLLocationManager *locationManager;

}                                                     ❷ 声明新成员变量以取得位置

- (void)locationManager:(CLLocationManager *)manager
     didUpdateToLocation:(CLLocation *)newLocation
➥fromLocation:(CLLocation *)oldLocation;

- (void)locationManager:(CLLocationManager *)manager
     ➥didFailWithError:(NSError *)error;

@property (nonatomic, retain) CLLocationManager *locationManager;

@end
```

在这段代码中可以看到新的核心位置头文件❶，以及保存位置管理器对象的新成员变量❷。这个成员变量会保存一个指向位置管理器的指针，以便找到用户的位置。下一个代码清单显示了修改后的 AroundMyCityController.m，用于获取当前位置。

代码清单 6-6　修改后的 AroundMyCityController.m 代码

```
//
//  AroundMyCityViewController.m
```

```
//  AroundMyCity
//
//  Created by Murat Aktihanoglu on 2/19/10.
//

#import "AroundMyCityViewController.h"
@implementation AroundMyCityViewController

@synthesize locationManager;

/*
// The designated initializer. Override to perform setup
// that is required before the view is loaded.
- (id)initWithNibName:(NSString *)nibNameOrNil
   ➥bundle:(NSBundle *)nibBundleOrNil {
    if (self = [super initWithNibName:nibNameOrNil
   ➥bundle:nibBundleOrNil]) {
       // Custom initialization
    }
    return self;
}
*/

/*
// Implement loadView to create a view hierarchy programmatically, without
     using a nib.
- (void)loadView {
}
*/

// Implement viewDidLoad to do additional setup
// after loading the view, typically from a nib.
- (void)viewDidLoad {
    [super viewDidLoad];

    locationManager=[[CLLocationManager alloc] init];
    locationManager.delegate=self;
    locationManager.desiredAccuracy=kCLLocationAccuracyNearestTenMeters;

    [locationManager startUpdatingLocation];

}


/*
// Override to allow orientations other than
// the default portrait orientation.
- (BOOL)shouldAutorotateToInterfaceOrientation:
  ➥(UIInterfaceOrientation)interfaceOrientation {
    // Return YES for supported orientations
    return (interfaceOrientation == UIInterfaceOrientationPortrait);
}
*/
```

实例化位置管理器

实现 viewDidLoad 函数

```
- (void)didReceiveMemoryWarning {
    // Releases the view if it doesn't have a superview.
    [super didReceiveMemoryWarning];

    // Release any cached data, images, etc that aren't in use.
}

- (void)viewDidUnload {
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}

- (void)dealloc {
    [super dealloc];
}

- (void)locationManager:(CLLocationManager *)manager
     didUpdateToLocation:(CLLocation *)newLocation
➥fromLocation:(CLLocation *)oldLocation{
    NSLog(@"update received!");
}

- (void)locationManager:(CLLocationManager *)manager
 didFailWithError:(NSError *)error{

    NSLog(@"ERROR on location");
}

@end
```

实现 didUpdateToLocation 函数

实现 didFailWithError 函数

6

这段修改后的代码能够从平台获取设备的位置信息。加载完视图后，代码就会初始化位置管理器，按要求获取位置。如果成功，异步程序就会调用 didUpdateToLocation 函数，否则就会调用 didFailWithError。

完成所有步骤后，选择 Run >Debug, iPhone 模拟器就会启动，测试该应用。

加载视图后就会调用 viewDidLoad 函数，检索位置。平台会多次而不是只调用一次该应用，位置更新也会越来越精确，以控制电池消耗和位置修正的时间。

6.5.3 Apple App Store

应用开发完后就要送到 Apple App Store（图 6-7）接受审核。通常 Apple 会用一到两周的时间决定接受或拒绝你的应用。如果接受了，就可以在 App Store 中列出来，而 Apple 就会托管你的应用程序供用户下载使用。如果用户下载该应用需要付费的话，Apple 就会拿走酬金的 30%。更多细节请参照表 6-3 和第 11 章。



图 6-7　苹果应用商店

6.6　Android

Android 是基于 Linux 内核的操作系统。最初由 Google 开发，后来由 Open Handset Alliance[①]（开放手机联盟）维护。这个系统允许开发人员用 Java 语言编写托管代码，通过 Google 研发的 Java 程序库来控制设备。图 6-8 展示了 Android 操作系统的主要架构。

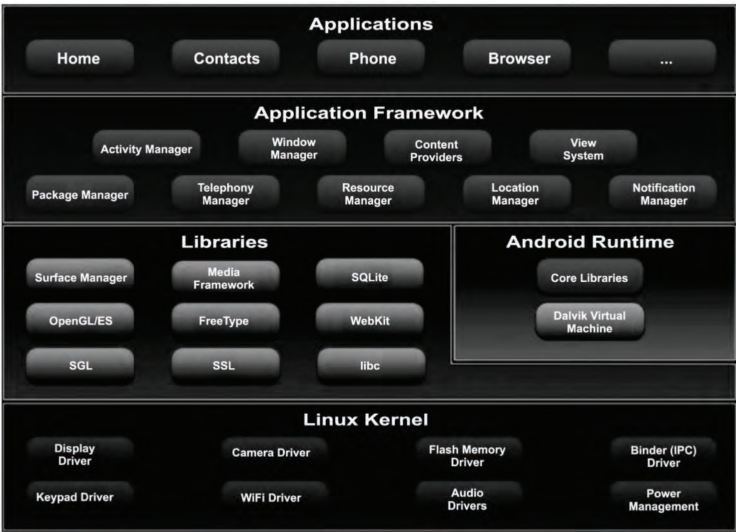


图 6-8　Android 操作系统的主要组成部分

① http://www.openhandsetalliance.com/

简单来说：Android 应用是使用 Java 通过 Android Java libraries[1]开发的，IDE 推荐 Eclipse（一个开源开发环境），应用可以提交给 Android Market。Android Market 不需要审批，不像 Apple App Store，提交给 Android Market 的应用可以立即可见。

Google 创建了 Android 这样功能强大又稳定的移动操作系统。Android 的开放性给较为古老的 OS 甚至 iPhone 带来巨大的冲击。很多设备制造商都从老旧难以维护的平台转到 Android 上来，发布新的 Android 手机。

比较著名的例子是 Motorola 放弃 Symbian 转向 Android，与 Verizon 合作开发出 Droid 手机直接与 iPhone 抗衡。Sony Ericsson 也选择 Android 操作系统，放弃了目前使用的 OSE、Symbian 和 Windows Mobile 平台。

随着 Android 2.0 的发布，Google 也推出了操作系统内置的免费导航应用，给需要付费的 TomTom 和 Garmin 应用带来不小冲击，从而使导航应用变得更普及。

学习 Android 开发很容易，Java 开发人员可以很快上手。Android Java 程序库非常直观，有助于将这个平台发展为开放的资源平台。实际上，如果代码运行不好，可以在操作系统程序库中做些调试。下面是 Android 平台的一些主要特色：

- 小部分 Android 设备具有多点触摸功能——说来奇怪，不像 iPhone 手机，Android SDK 一直忽略了触摸功能，直到最近才推出支持此功能的版本。Google 声称很快会推广这项功能，因为一些开发人员通过破解 OS，在 Android 手机上创建了多点触摸示例应用。
- Multitasking（多任务功能）——Android 支持多任务，不过需要做一点说明。在 Windows 桌面上，你可以随便启动、关闭程序或在运行的程序之间切换。不管这个程序是否是焦点或在后台，它们仍以相同的方式运行着。在 Android 中，你不需要关闭程序，因为该系统会在内存即将用尽时自动替你关掉它。

不像 Windows，Android 中正在运行的程序（比如下载文件或播放音乐）和不用的程序之间区分很清楚。一般所说的“Android 可以在后台运行程序”指后台程序正在运行任务。例如，你可以在浏览邮件时下载文件。

有一种情况很难定义，就是打开一个程序但没有使用的情况。在这种条件下，该程序会保持运行状态，直到系统需要内存时停止；那么它就会在不提醒的情况下被关闭。正因为这样，应用开发人员可以制造一个流畅的多任务处理的假象。

也就是说，每次用户不再关注一种应用（为了在屏幕上显示另外一个程序，该应用就不再显示）时，必须将其状态保存到持久存储器上。否则，就会在系统重新分配内存时重置该应用。因为此应用在关闭前无法知道是否会被重新启用，唯一恰当的解决方法就是把内存中的每一个变量都保存下来，然后在重新启用时重新装载这些变量。有鉴于此，可以说 Android 并不真正支持多任务，而只是制造了一个假象而已。

结论本身有点容易让人误解，因为 Android 的确支持多任务，可是它又要求所有规范的程序都必须在失去焦点后假设自己将会被关闭，结果就和不支持多任务没有多大的区别了。其真

① http://code.google.com/android/download_list.html

正的优势在于用户在重新关注该程序前，它并不会关闭，那么就会节省从永久存储器中调用相同状态程序的一些时间。

6.6.1　简单的Android 应用

现在让我们开发一个简单的 Android 应用，同样是找到用户的当前位置。示例代码会访问平台的 Location Manager 服务，由此获取用户位置。步骤如下：

(1) 为下载安装 Eclipse IDE for Java Developers[①]。

(2) 在 IDE 中安装 Android Development Toolkit。这就把 Eclipse 变成了 Android 的优秀开发平台了，强烈推荐这种配置。为此，在 Eclipse 中选择 Help > Install New Software，添加 https://dl-ssl.google.com/android/eclipse/站点。接着就会显示可用的 Developer Tools。选择并安装 ADT。现在就已经准备好了带有 Android 插件的 Eclipse 了。

(3) 从 http://developer.android.com 上下载安装最新版的 Android SDK。

(4) 在 Eclipse 中，选择 Window > Preferences，在 Android 选项卡中指定 Android SDK 所在的位置。这样就会将 Eclipse 开发平台与 Android SDK 的位置联系起来。

(5) 选择 File > New Project >Android，填写项目和应用名称，在 Eclipse 中创建一个新 Android 项目。

代码清单 6-7　自动生成的 Android 应用源代码

```
package LBSBook.AroundMyCity;

import android.app.Activity;
import android.os.Bundle;

public class AroundMyCity extends Activity {        ❶ 声明主活动类

    @Override
    public void onCreate(Bundle savedInstanceState) {        ❷ 创建活动时调用

        super.onCreate(savedInstanceState);        ❸ 调用 onCreate 的父类实现
        setContentView(R.layout.main);
    }
}
```

从这段代码中可以看到应用的主要活动类❶及其成员函数❷。自定义 onCreate 函数首先调用父级类的默认实现❸。

(6) 打开 AndroidManifest.xml 文件，为此应用添加访问 Location Manager 系统组件的许可。添加下面的文件：

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

① http://eclipse.org

提示 在 Android 上访问位置要获得授权注意 ACCESS_COARSE_LOCATION 允许应用程序访问低精度（比如 Cell ID、Wi-Fi）位置，ACCESS_FINE_LOCATION 允许应用程序访问精确（比如 GPS）位置。

(7) 导入 Location 和 LocationManager 类。

(8) 从系统中获取 Location Manager，调用它从而得到当前的位置。修改 AroundMyCity.java，其结果应像下面代码清单中所示。

代码清单 6-8 改进的 AroundMyCity.java 源代码

```
package com.LBSBook;

import android.app.Activity;
import android.os.Bundle;
import android.content.Context;
import android.widget.TextView;
import android.location.Location;
import android.location.LocationManager;

public class HelloAndroid extends Activity {          <- 声明主活动类

    @Override
    public void onCreate(Bundle savedInstanceState) {  <- 实例化时调用

        super.onCreate(savedInstanceState);            <- 调用 onCreate 的父类实现

        LocationManager locationManager =
➥(LocationManager)getSystemService
➥(Context.LOCATION_SERVICE);                           <- ❶ 取得 LocationManager

        Location location =
➥locationManager.getLastKnownLocation("gps");          <- ❷ 取得上次已知的位置

        TextView tv = new TextView(this);
        tv.setText("Location is " + location.getLatitude()
➥+ ", " + location.getLongitude() );
        setContentView(tv);                            <- ❸ 用位置值更新用户界面

    }
}
```

从这段代码中可以看到从平台中调出 Location Manager❶需要的修改，以及怎样从设备中获取位置❷，相应地更新 UI❸。

所有步骤完成后，点击 Debug，选择 Android Application，这样就能安装此应用，并在 USB 连接着的 Android 手机上运行。

在前面介绍的代码清单中，主要应用类扩展了 `Activity` 类。只要启动活动，就会调用 `onCreate` 函数。代码首先调用父类的 onCreate 函数，然后获取用户位置并创建一个新的 TextView

对象，为其设置好文本及内容视图，以便在 TextView 屏幕上显示其内容。

6.6.2　Android Market

Android Market（图 6-9）是由 Google 为 Android 系统用户创建的一种软件应用服务，用户可以从 Android Market 浏览和下载第三方开发商发行的应用程序。它在 2008 年 8 月 28 日推出，在 2008 年 10 月 22 日可以供用户使用。在 2009 年 12 月中旬为美国和英国的用户添加了可收费的应用支持。所有递交的申请可以立即生效，这与其他的应用商店截然不同。这使漏洞修复的发布更便捷。但是如果你开发的应用产品不够有特色，在 Android Market 中就很难被用户发现。更多细节请参看表 6-3 和第 11 章。

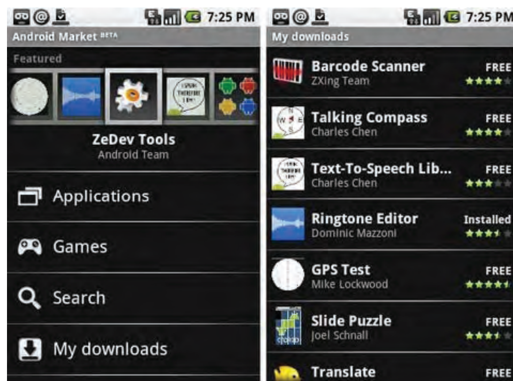


图 6-9　Android Market

6.7　webOS

webOS 是 Linux 内核的智能手机平台，由 Palm 开发，现在被 HP 收归旗下。该平台于 2009 年 1 月 8 日的拉斯维加斯国际消费电子展上展出，并于 2009 年 6 月 6 日发布。第一款搭载 webOS 系统的智能手机是 Palm Pre，于 2009 年 6 月 6 日发售。使用此操作系统的第二款设备是 Palm Pi。webOS 将在线社交网络和 Web 2.0 一体化作为重点。要了解 WebOS 架构概况请看图 6-10。

webOS 的图形用户界面是为触摸屏设备设计的。它包含了一系列个人信息管理应用，采用 HTML 5、JavaScript 和 CSS 等技术。Palm 声称这种基于现有技术的设计是为了避免开发人员学习新的编程语言。2009 年 6 月 6 日发售的第一款搭载此平台的设备就是 The Palm Pre。

由于 webOS 基于广为人知的技术，所以很容易上手。但是开发人员仍需要学习熟悉 webOS 应用的总体结构。每一种 webOS 应用都有一个场景组件。而每个场景都是为了呈现用户信息或任务的格式化屏幕，都有一个视图和一个助手。视图决定了这个场景的布局和外观，助手决定了行为。有些场景也具提供数据的模块。

webOS 是一个真正的多任务操作系统。用户可以在运行的应用之间相互切换。就像 iPhone 那样，它也支持多点触摸功能。

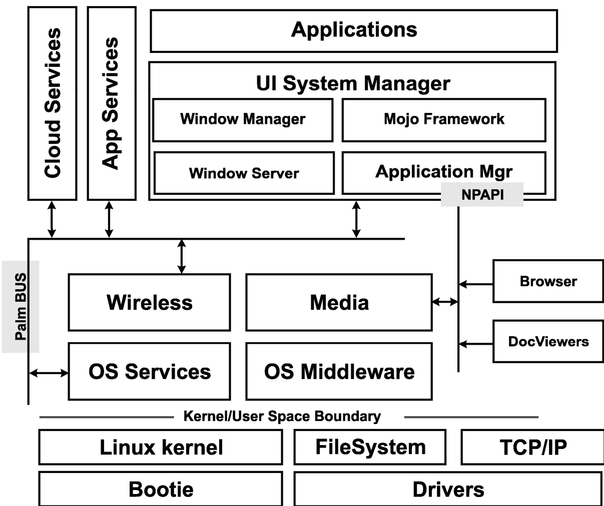


图 6-10 webOS 结构（来源：mng.bz/E16W）

Palm 一直在努力说服开发人员使用其平台，虽然其前景尚不明朗，但这的确是一个革命性的平台。

HP App Catalog

HP App Catalog（图 6-11）是为运行 webOS 平台的 Palm 移动设备提供应用程序的在线市场，但是不包括 Palm OS 用户。最初，只有 Palm Pre 才能下载使用这些应用。

HP App Catalog 与 App Store 以及 Android Marketplace 相似。在 Palm Pre 手机上市时，App Catalog 有 18 种应用。这些应用必须通过 Developer Portal 认证后才能提交给 App Catalog。更多信息请看表 6-3 和第 11 章。



图 6-11 HP App Catalog

6.8　BlackBerry OS

BlackBerry OS 是 RIM（Research In Motion）为其智能手机产品BlackBerry开发的专用操作系统。这一操作系统具有多任务处理能力，并支持特定的输入装置，如轨迹球或触摸屏。

BlackBerry 手机开发使用 Java ME，Java ME 本身是 Java 1.3 的衍生版（不是 1.5 版，1.5 版具有泛型、枚举等最新的功能）。在应用提交前（直接或通过 BlackBerry App World）必须通过加密签名获得 RIM 的安全密钥。BlackBerry 平台如图 6-12 所示。

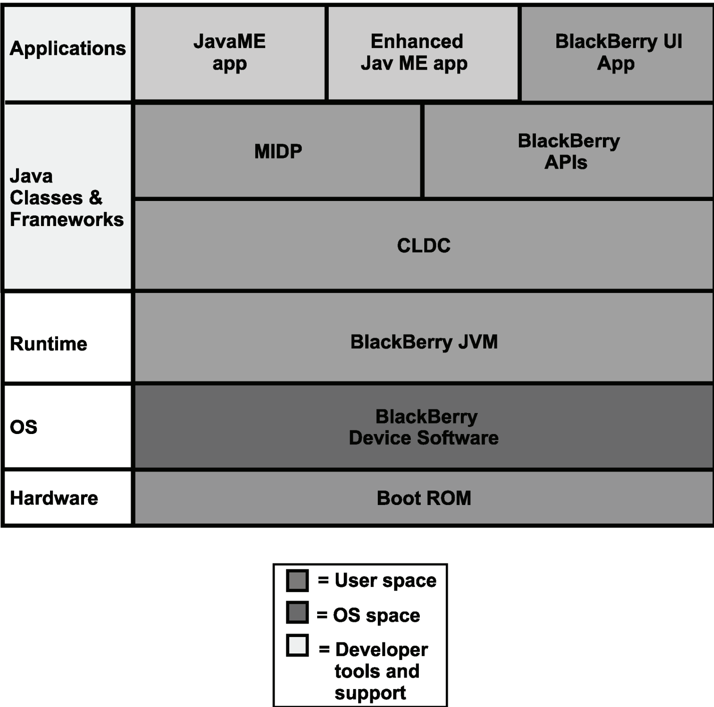


图 6-12　BlackBerry 结构（来源：mng.bz/1hue）

BlackBerry 平台基于 CLDC[1]开发的，也支持 MIDP2.0[2]。

- CLDC——The Connected Limited Device Configuration（有限连接设备配置，CLDC）是为资源受限的设备，比如手机、寻呼机、个人数码助理，制订的有关应用程序接口和虚拟机的基本架构。再加上 Mobile Information Device Profile（MIDP，移动信息设备配置文件），就能为应用开发提供一个完善的 Java 平台，开发能在有限内存、处理能力和绘图能力的设备上运行的应用。

① http://java.sun.com/products/cldc/
② http://java.sun.com/products/midp/

- MIDP——让你能够为连网的移动设备编写可下载的应用和服务。与 CLDC 结合，MIDP 就能为当下最流行的小型移动信息设备，比如手机和主流 PDA，提供 Java 运行环境。

BlackBerry 平台同样有大批专有类。这就让人犹豫是写 BlackBerry 应用还是写 MIDP 应用。

对于 MIDP 应用来说，开发人员必须只使用由 CLDC 和 MIDP 标准公布的 API。这样的应用可以在 BlackBerry 和多数的手机上运行，但是不能运用任何 BlackBerry 特定的功能。你可以利用任何一种与 MIDP 兼容的开发工具，比如 CLDC 的 Sun Java Wireless Toolkit（以前称作 J2ME Wireless Toolkit），来开发应用程序。

多数应用开发人员选择搭建 BlackBerry 特定应用。这就意味着需要学习包括新用户界面类在内的 BlackBerry 专有 APIs，以及利用 BlackBerry Java Development Environment (JDE)[①]来开发应用。JDE 可以免费下载，适用于标准的 Java Software Development Kit，可以为用户提供创建、打包、测试及调试 BlackBerry 应用所需的所有工具。你甚至不需要 BlackBerry 手机，因为 JDE 包含了具有 BlackBerry 所有功能的模拟器。它能向编程人员提供一整套类和接口的文档。

BlackBerry App World

BlackBerry App World（图 6-13）是一种应用分发服务，是 RIM 为某些 BlackBerry 设备创建的一种应用。在这种环境下，BlackBerry App 用户可以浏览、下载和更新由第三方开发的应用程序。BlackBerry App World 于 2009 年 4 月 1 日创立，其更多信息请参看表 6-3 和第 11 章。

6

提示 BlackBerry 手机上的媒体卡不能存储应用程序，所以每次能下载到设备上使用的应用非常有限。

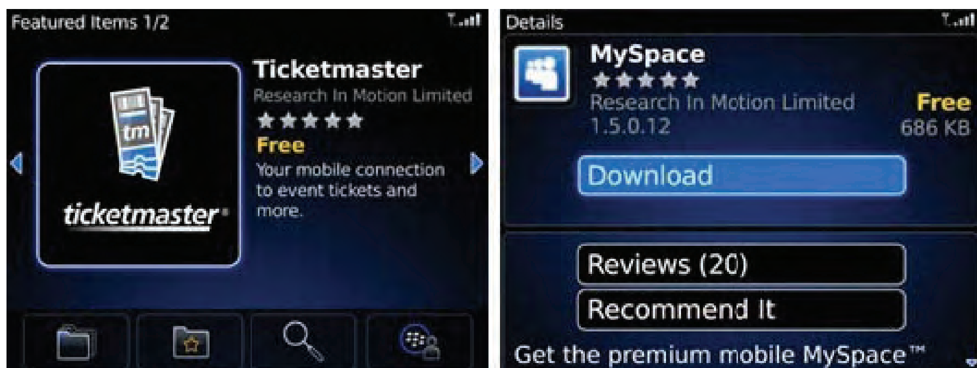


图 6-13 BlackBerry App World

① http://supportforums.blackberry.com/t5/Java-Development/tkb-p/java_dev@tkb

6.9　Windows Mobile

Windows Mobile 是一种基于 Microsoft Win32 API 为移动设备开发的整合了一系列基础应用的简洁操作系统。使用 Windows Mobile 操作系统的设备主要有 Pocket PC、智能手机、Portable Media Centers（随身音乐播放器）、一些车载电脑等。

Windows Mobile 最初是 Pocket PC 2000 操作系统，至今已被更新数次，上一个版本 Windows Mobile 6.5 是在 2009 年秋天发布的。Windows Phone 7 是 2010 年 12 月 7 日发布的最新版本，2010 年 9 月 16 日它的 SDK 发布。

编写能运行在 Windows Mobile 手机上的程序与为 Windows 台式机编写程序非常相似。可以使用 Visual Studio 等工具，可以在 Windows Marketplace for Mobile（微软手机应用程序商店）上发行应用程序。

微软手机应用商店

Windows Marketplace for Mobile，即微软手机应用商店（图 6-14）是由微软为其 Windows Mobile 平台研发的一种应用和服务。用户可以在这里浏览和下载由第三方开发的应用程序。目前，此应用可以在 Windows Mobile 6.5 设备以及个人电脑上直接使用。Windows Marketplace for Mobile 在 2009 年世界移动大会上宣布，在 2009 年 10 月 6 日推出。更多信息请参看表 6-3 和第 11 章。



图 6-14　微软手机应用程序商店

Windows Marketplace for Mobile 对 Windows 手机和个人电脑开放，向买主提供 24 小时退货政策，将每份应用收入的 70%支付给开发人员。微软公司每年向列出 5 个应用程序的开发人员一次性收取 99 美元的费用。除已列出的 5 个应用程序之外，对每个额外的应用只能另外收取每年 99 美元的费用。更多信息请参看表 6-3。

6.10 LiMo

The LiMo（Linux Mobile）Foundation（LiMo 基金会）是由 Motorola、NEC、NTT DoCoMo、Panasonic Mobile Communications、Samsung Electronics 和 Vodafone 在 2007 年 1 月建立的全球联盟，意在为移动设备建立一个基于 Linux 的全球竞争的操作系统。至今，该联盟的会员数超过 50，他们在一个开放透明的管理模式内彼此合作，采用共享式领导与决策，以便交付一种开放、全球一致、基于 Mobile Linux 的手机软件平台，供整个移动行业使用。图 6-15 展示了 LiMo 架构的概况。

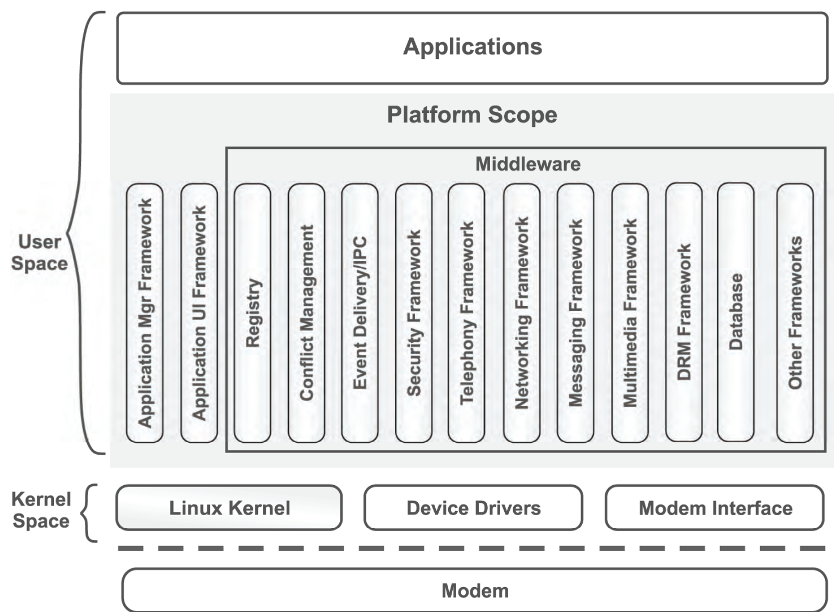


图 6-15 LiMo 结构（来源：mng.bz/t946）

它具有模块式插件程序架构，支持 DRM。LiMo 应用程序开发人员可以利用 SDK 编写能在 Java 虚拟机上运行的托管代码、基于 WebKit 的浏览器应用程序以及本地代码。正如在图 6-15 上所看到的，这个平台能在带有中间件的 Linux 内核上运行，这些中间件可以提供基本的功能。

Orange 和 Access 已经许可 LiMo 开发蜂窝手机，但目前 LiMo 是否将会成为值得开发的主要平台还不能确定。

6.11 MeeGo

英特尔开发了 Moblin 平台，诺基亚开发了 Maemo，2010 年 12 月这两个平台被融合为一个称作 MeeGo。MeeGo 是基于 Moblin 内核的操作系统，以 QT 作为应用开发环境。通过 Linux Foundation 的支持，它将是一个完全开放的平台，因此每个人都能参与。它能在笔记本、上网本、

平板电脑、音乐手机、在线电视、车辆信息娱乐系统及其他设备上使用。我们快速看一下 Moblin 和 Maemo 以更好地了解 MeeGo 的基本情况。

Moblin 是 Mobile Linux 的简称，它是一个开放源操作系统，也是为移动因特网设备（MIDs）、上网本和上网机开发的一种应用套件。基于 Intel Atom 处理器，Moblin 目前的架构就是为了减少启动时间和电池消耗，创建以上网本和 MID 为核心的操作系统。Moblin 架构概况如图 6-16 所示。

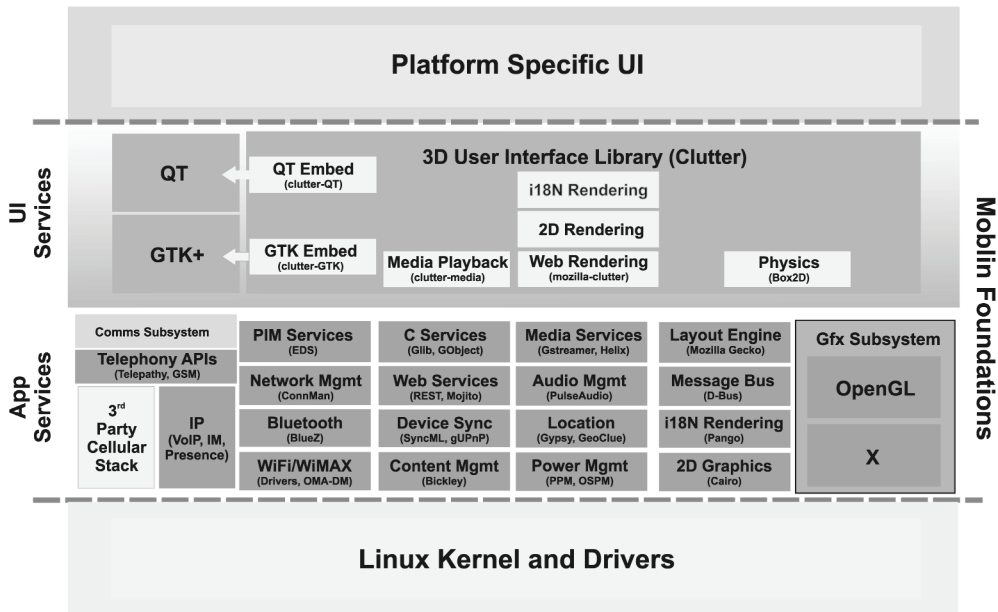


图 6-16　Moblin 架构（来源：mng.bz/XRT6）

Moblin 具有 Linux 内核，带有 App 和 UI（用户界面）服务层，还有一个专门为智能手机和上网本等移动设备开发的特定 UI 平台。

Intel 在 2007 年 7 月推出了 Moblin.org 的网站，2008 年 4 月在上海召开的英特尔发展论坛上，Intel 隆重推出了 Intel Atom 处理器系列，彻底更新了该网站。在该网站上还能自定义 SDK。其中 Moblin 2 OS 是专门为 Intel Atom 处理器的上网本设计研发的。2009 年 4 月 2 日 Intel 把 Moblin 操作系统移交给 Linux 基金会。为 Moblin 2 量身打造的商业产品包括 Foxconn 上网本和 InvenTech 智能手机。

Maemo 是由诺基亚为智能手机和平板电脑开发的一种软件平台。基于此平台开发的第一款手机是 Nokia N900。

Maemo 是针对移动设备的 Debian Linux 的修改版，在 Debian Linux 基础上做了一定的缩减。该平台采用了基于 X Window System 的用户界面，以 Xomap 和 Matchbox 作为窗口管理器；GUI 使用的是 GTK+工具箱、Hildon 用户界面小部件和 API。

虽然 Maemo 是基于 Linux 的开源软件，其中有些部分仍是不对外公开的。有些用户空间软

件，比如某些状态栏、工具栏小程序（包括显示器亮度调节程序）和应用，一些有关连通性和电源管理的后台程序系统，不会像其他系统那样对外公开。

Maemo 应用可以通过 Maemo SDK，用 C 语言开发，也可以用 Java 语言来开发，在 Jalimo JVM 上运行，或者用 Python、 Ruby 和 Mono 开发。

诺基亚弃用塞班平台而选择了 Maemo 开发其 N 系列高端手机，因此 Maemo 有可能变得越来越重要。另外，Maemo 应用程序也能在 Nokia 的 Ovi Store 上下载。

6.12 BREW

BREW（Binary Run time Environment for Wireless，无线二进制运行环境）在 2001 年 9 月首次公布，它是由 Qualcomm（高通公司）为手机创建的一款应用开发平台。起初是为 CDMA 网络手机研发的，但是后来就转移到包括 GSM/GPRS 在内的其他平台上。

BREW 是一个软件平台，能够下载和运行游戏、短信、相册等小程序。BREW 平台最主要的优势，就是应用程序开发人员可以将他们的应用在所有高通公司的设备之间移植。它的作用就是为应用和无线设备芯片操作系统架起桥梁，让编程人员无须编写系统程序或是了解无线应用就可以开发应用。图 6-17 是 BREW 架构的概况。

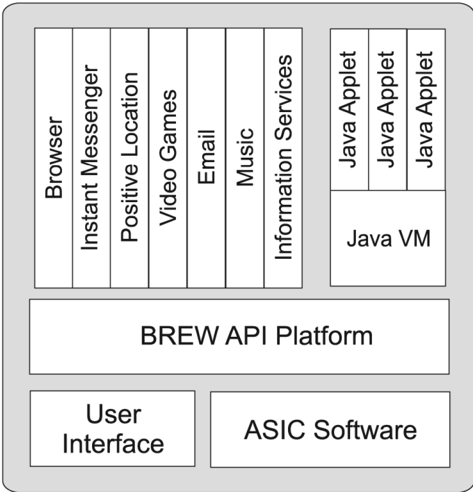


图 6-17 BREW 架构（来源：mng.bz/S0aX）

BREW 开发通常耗时较长，运营商需要核证应用程序（这可能需要最长两周的测试时间）。正因为如此， BREW 开发需要较高的费用。写完一个应用，每次 True BREW 测试需要两周的时间。接下来，就需要与运营商协商。然后（如果协商成功）运营商会花费时间用自己的网络重新测试该应用。最后，如果要推出新的版本，还得重复这个过程。

提示　**BREW 与 JAVA ME**　Java ME 在欧洲使用很广，而 BRFW 主要用在美国和日本。即使是在美国，Java ME 手机曾经占据的市场份额也比 BRFW 手机多。

现在的商业技术将Java ME应用移植到BREW，这样对同种应用就不需要用Java和C/C++两种语言开发两个版本，降低了门槛。

创建一个应用有时可能不需要从零开始，有些应用可以利用移动开发框架来创建。

6.13　移动开发框架

可以使用框架来创建移动应用程序，甚至无须编写任何代码。这些框架在功能上有非常明确的限制条件，建议了解这些框架能做什么不能做什么。现在看一下一些可用的移动开发框架。

6.13.1　PhoneGap

PhoneGap[1]是一个开源的开发工具，可以使用 JavaScript 快速、简单地开发移动应用程序。Web 开发人员可以用 HTML 和 JavaScript 来创建移动应用，而仍能利用 iPhone、Android 和 Blackberry SDK 的核心功能。表 6-5 列举了不同平台中 PhoneGap 可用的功能。

表6-5　不同平台中PhoneGap功能

	iPhone	Android	Blackberry
地理位置	是	是	是
振动	是	是	是
加速计	是	是	暂无
声音	是	是	是
支持	是	暂无	是

6.13.2　Kyte 移动应用程序框架

Kyte Mobile App Frameworks[2]是一个整体解决方案，Kyte 合作伙伴可以创建包含视频、在线聊天以及带有支付选项的应用，且开发成本不高。

6.13.3　Big5

Big5[3]能使你的网络应用程序访问加速度计数据以获取当前的地理位置，还利用内置照相机及更多工具。这些都与 PhoneGap 相似，但是所有的应用只能在 Big5Apps 容器中运行。

① http://www.phonegap.com/
② http://www.kyte.com/platform/pg/kyte_mobile_app_frameworks
③ http://www.big5apps.com/

6.13.4 Titanium Mobile

Titanium Mobile[1]是一个与 PhoneGap 相似的工具箱，可以允许 Web 应用程序访问平台 API。

6.13.5 QuickConnect

QuickConnect[2]是一个功能强大、模块化、使用简单的应用开发程序库，可用于很多语言和平台。QuickConnect 目前可用于 iPhone、 Android、MacJavaScript apps、Erlang/Yaws 和 PHP。

6.13.6 Rhodes 框架

Rhodes[3]是一个开源框架，能与本地数据同步，利用诸如 GPS、PIM contacts 和照相机这样的设备功能为很多智能手机操作系统（iPhone、Windows Mobile、RIM、Symbian 和 Android）快速构建本地应用（而非移动 Web 应用）。

每一种移动应用在提交给应用商店之前都要经过反复测试，对开发人员来说，有很多种方法可以降低测试成本。

6.14 测试

在每一个目标设备上测试移动应用是开发过程中很重要的一部分。因为每个设备对你采用的 API 都有不同的底层实现，所以在每个目标设备上都必须进行测试。如果你开发的是 Java ME 应用程序，就不得不在成百上千的不同硬件平台上进行测试。如果你没有上百部手机，想要测试是不切实际的。

像 DeviceAnywhere[4]和 Intertek[5]这样的服务可以让开发者远程访问很多的手机，这些手机上安装和测试应用程序。例如，DeviceAnywhere 就能通过因特网提供超过 2000 部的移动设备，以供应用测试使用。

Mob4Hire[6]可以实现移动应用程序的群测。Mob4Hire 网站将全球的测试人员和需要在多种手机和多种操作系统上测试其应用程序的开发人员联系起来。Mob4Hire 会拿走 15%的费用，PayPal 通常收取 2.9%的服务费，外加每笔交易 30 美分的费用。得到报酬的测试人员就会在他们的手机上运行该应用，报告他们所发现的问题。这种测试方法的所有费用比成立公司组织人员在不同地区测试要少得多。

---

① http://www.appcelerator.com/
② http://sourceforge.net/projects/quickconnect/
③ http://rhomobile.com/
④ http://www.deviceanywhere.com/
⑤ http://www.intertek.com/wireless-mobile/
⑥ http://www.mob4hire.com/

6.15 小结

苹果公司和谷歌的 iPhone 和 Android 彻底改变了移动开发产业。运营商不再握有权力，这种权力转移到了手机制造商和开发者的手中。

目前最好的 LBS 平台就是 iPhone 和 Android。不像其他的平台需要基站数据库支持，这两个平台能够很容易实现设备位置检索。RIM 也发布了一些新平台 API 能够支持 GPS，但还没有被广泛采用。

Android 是最有前景的平台，几乎每个手机制造商都生产很多这种平台的新设备，但是 iPhone 的出现对 Android 造成了冲击。

另外，用 HTML5 编写基于浏览器的 LBS 应用程序是非常有前景的，因为它也能在移动浏览器上运行。

目前的应用商店仍是开发者最头疼的问题。过去曾由运营商来掌控批准大权，现在改由应用商店来进行审批。给不同的应用商店开发应用，就必须按其要求制订相应的计划。你的应用可能会遭拒，或是审批流程实在太长，不适合更新升级。

应用商店的第二个主要问题就是可见度问题。目前，不是所有的应用商店都实现了智能过滤和推荐功能。在只能显示前 50 个程序的商店里，后面很多长尾程序卖得不会太好，下载量也不会太大。

一般情况下，Java ME 应用应该能在任何一款手机上运行，但实际上并非如此。你需要在所有的目标设备上测试应用程序。

第 7 章

连接问题

本章内容

- ❑ 连接成功的关键因素
- ❑ 位置数据安全
- ❑ 位置感知应用实例

毋庸置疑，使用 LBS 的一个关键好处就是能够自动检测移动用户所在的位置，并能提供基于用户位置的相关信息。不断获知用户位置的要求引发了很多的开发难题。这些问题的解决方法应该能预见用户可能出现的反应。

如果是在一个特定的较短周期内使用应用，那么“随用随连”将会比“实时在线”应用表现得更好，但前者很有可能会比后者花费用户更多的时间来获得位置信息。另外，手机的内存限制也要求开发人员采用一种巧妙的策略，在确保用户能快速访问位置信息的同时，又尽可能地简化其操作。

本章中，我们会分析成功 LBS 应用的关键连接因素，比如在位置移动时快速锁定方位、恰当的定位精度、最小的电量耗费，以及在移动时对用户连续发出有价值的反馈信息。

7.1 连接中的关键成功因素

目前的 GPS 平台，比如智能手机，通常比使用相同技术的专用个人导航装置表现得更差。智能手机定位能力落后是由于电池消耗大、GPS 卫星信号锁定位置有时间要求（因为 GPS 设备为了省电只能间歇性开机），以及内置于设备中的其他器件的电子干扰造成定位不准或错误。而个人导航装置作为单一功能的设备还备有更好的信号接收天线，因此可以更有效地利用电源。

对基于位置的移动应用来说，想要顺利且有效地连接，下面是一些重要的影响因素：

- ❑ 快速锁定位置——由于 GPS 定位需要花费很长的时间，有可能需要几分钟甚至超过 10 分钟的 TIFF 时间(首次锁定位置的时间)来首次确定位置，因此会对那些使用基于位置的无线网络服务的用户造成不便。
- ❑ 适当的精度——高精度的定位需要花费较长的时间和较多的电量。LBS 的服务性质决定了该应用会一直要求位置信息能满足足够的精度，同时要优化电量的使用并且要减少

TIFF 时间，如图 7-1 所示。

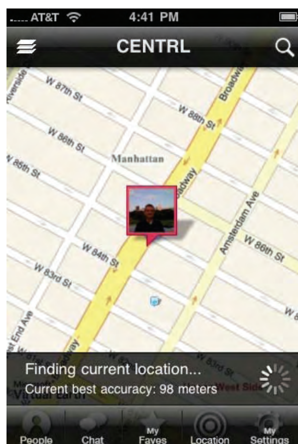


图 7-1　LBS 应用实时显示用户定位的精度

- 减少电池消耗——一直使用 GPS 功能会耗尽电池电量。GPS 并不是以导航为主的设备，如果想一直获取有效的位置就会很快地耗尽电量。
- 连续适当的信息反馈——你开发的应用应该能保障用户一直知道其位置信息。不管正在做什么，都应该通过用户界面不断地向用户反馈信息。在定位时没有响应的应用设计是不能忍受的。

7.1.1　智能手机

iPhone[1]、Android[2]、BlackBerry[3]、Web OS[4]，以及 Windows Phone 7[5]等智能手机平台通过提供简单的原生 API 来检索设备位置，改变了移动应用开发的现状。研发人员不再需要使用定制的数据库和算法自己去查了。这种新方法导致智能手机平台掀起了一场基于位置应用的革命。

要开发一个智能手机上的 LBS 应用，如图 7-2 所示，你要做的全部工作就是调用 API 获取用户的位置信息。但其中仍有很多注意事项：

- 不要过于频繁地获取用户位置信息。
- 调整精度以满足应用所能接受的最低限度。
- 在用户界面上经常告知用户现在的进度。

现在让我们了解一下为功能手机开发位置感知应用的一些技巧。

① http://developer.apple.com/iphone/
② http://developer.android.com/index.html
③ http://us.blackberry.com/developers/
④ http://developer.palm.com/
⑤ mng.bz/9jIk

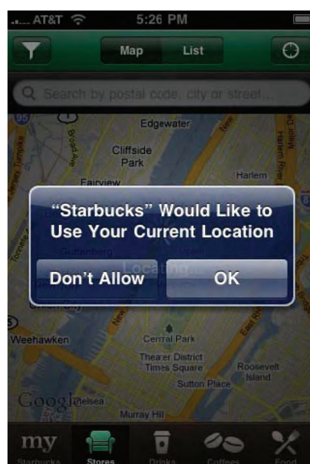


图 7-2　请求获取用户当前位置的 iPhone LBS 应用

7.1.2　功能手机

运行 Java ME 的手机要比前面提到的智能手机平台更有挑战性。

一些 Java ME 平台的手机内置了位置供应模块，可以提供设备的位置信息，如代码清单 7.3 所示。

如果 Java ME 平台没有内置定位功能，应用程序通常需要访问设备附近的基站，查询数据库找到这些基站的位置，然后通过某种算法估计出设备所在的位置。基站位置数据库有商用版，也有像 celldb.org 和 opencellid.org 这样的开源项目数据库。

即使对有位置提供程序的特定设备来说，通常也应该把 LBS 应用设计为在没有内置位置提供程序的状态下也可以运行。这样在简单 Java ME 平台上开发 LBS 应用就比在智能手机上开发更复杂，但是通过选择恰当的设计和编码可以有效地弱化这些不利因素。

下面我们会了解下位置数据的安全问题，对消费者来说安全是他们关注的另一个主要问题。

7.2　位置数据的安全性

电子前沿基金会（Electronic Frontier Foundation）[①]和其他不同的组织机构曾对隐私和 LBS 做过深入的调查，发现的一个明显的问题就是这项服务可能会将你的位置暴露给其他人。

任何 LBS 应用开发的第一步就是要非常明确地知道用户位置数据是如何使用的，保存在哪里（假如真的保存着）。有些 LBS 应用，比如 Centrl，并不保留用户位置历史信息。这些应用只是在一定时间段内（比如一周）保留最后一次所知的用户位置，再早的用户位置信息就从该系统中移除了。

① http://www.eff.org/

第二步就是一直让用户自己决定怎样更新他们的位置信息。通过一些设置，这可以很容易地实现，如图 7-3 所示。

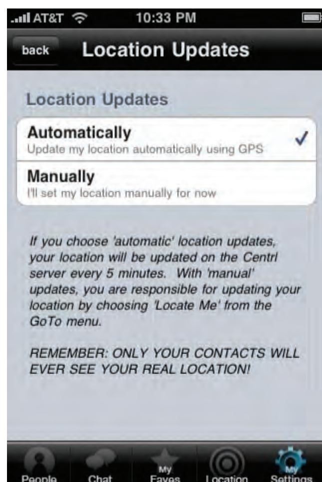


图 7-3　Centrl 的 LBS 应用中更新用户位置的设置

第三步就是确保该应用不会偶然地将用户位置及其历史信息暴露到这些连接到的移动设备上或者能记录这些数据的服务器上。

7.2.1　缓冲位置文件

LBS 应用必须确保第三方不能访问到任何存储在本地移动设备上的位置文件。这一点可以通过加密以及移除那些存储时间很长且已废弃的不必要的文件来轻松实现。

7.2.2　数据库服务器

LBS 供应商必须采用合理的管理方法、定期检查以及技术保障措施来保护用户的位置信息不被非法地访问使用、修改、破坏或披露。另外，必须经常对服务器进行检测和测试，以避免未被授权的个人访问该数据库。

现在让我们看一些代码实例，了解不同类型的平台是如何开始构建的。

7.3　位置感知平台实例

在不同平台上有各种 API 来检索用户位置。当前人们对 LBS 的关注促进了 LBS 底层技术的不断改进，使得这些 API 变得越来越稳定。

现在通过一些实例了解一下如何在一些平台上获取用户位置，先从 iPhone 开始。

7.3.1 iPhone和iPad实例

第一个代码清单列举的是在 iPhone 和 iPad[①]上获取用户位置的例子。

代码清单 7-1 在 iPhone 和 iPad 上使用 CoreLocation API

```
#import "MyLocationGetter.h"
#import <CoreLocation/CoreLocation.h>

@implementation MyLocationGetter

- (void)startUpdates
{
    if (nil == locationManager)
        locationManager = [[CLLocationManager alloc] init];

    locationManager.delegate = self;

    locationManager.distanceFilter = 1000;

    locationManager.desiredAccuracy = kCLLocationAccuracyKilometer;

    [locationManager startUpdatingLocation];

}


- (void)locationManager:(CLLocationManager *)manager
    didUpdateToLocation:(CLLocation *)newLocation
    fromLocation:(CLLocation *)oldLocation
{

    [manager stopUpdatingLocation];

     printf("latitude %+.6f, longitude %+.6f\n",
            newLocation.coordinate.latitude,
            newLocation.coordinate.longitude);

}
```

❶ 创建位置管理器
设置过滤器 1000 米
❷ 发出获取位置信息的请求
❸ 从协议代理方法
❹ 停止更新以省电
❺ 打印位置

这段代码显示了如果没有位置管理器，要怎样创建一个位置管理器❶，然后发出获取位置信息的请求❷。通过 `didUpdateToLocation` 函数获取位置值❸，停止更新位置请求以省电❹，然后显示检索到的位置值❺。

现在看看怎样在 Android 平台上完成同样的任务。

7.3.2 Android实例

下面的代码清单展示了在 Android[②]设备上获取位置的例子。

---

① http://developer.apple.com/iphone/

② http://developer.android.com/index.html

代码清单 7-2　在 Android 上使用位置 API

```
public class helloworld extends Activity {          ◁── ❶ 声明主活动类

    //@Override

    public TextView tv = null;
    public LocationManager locationManager = null;

    public void onCreate(Bundle icicle) {          ◁┐
                                                     ❷ 创建活动时调用
       super.onCreate(icicle);

       this.tv = new TextView(this);

       this.locationManager =
       (LocationManager)getSystemService(LOCATION_SERVICE);   ◁┐
                                                                ❸ 从系统获取位置
       String LOCATION_CHANGED = "location changed";              管理器

       IntentFilter filter = new IntentFilter(LOCATION_CHANGED);

       myIntentReceiver receiver =
       new myIntentReceiver();          ◁┐
                                          ❹ 创建新的 Intent 接收器
       registerReceiver(receiver, filter);

       List<LocationProvider> providers = locationManager.getProviders();

       LocationProvider gpsprovider = providers.get(0);
     Intent intent = new Intent(LOCATION_CHANGED);

     locationManager.requestUpdates(gpsprovider,   ◁┐
        0, 0, intent);                               ❺ 向位置管理器请求更新

     setContentView(tv);
  }                                        ❻ 取回位置值后调用

  public void updateLocation() {          ◁┘

  URL urlConn = new URL("http://www.google.com/");

     URLConnection httpConn = urlConn.openConnection();
                                                       ❼ 获得位置值
     Location location =
 this.locationManager.getCurrentLocation("gps");          ◁┘

        this.tv.setText("Hello, World. "
           + location.convert(location.getLatitude(),
             ➥location.FORMAT_DEGREES)          ◁┐
 + ", "                                             ❽ 显示取回的位置值
          location.convert(location.getLongitude(),
```

```
            ➥location.FORMAT_DEGREES));
        }
    }
```

这段代码有个主要的活动类❶能够从系统中❸获取位置管理器❷。它创建了一个 Intent ❹，然后开始从中请求进行位置更新❺。一旦取回位置值，就可以调用 updateLocation❻函数。这个函数就能得到最新的位置值❼，然后在用户界面上将其显示出来❽。

现在看看在一般的 Java ME 平台上如何完成相同的任务。

7.3.3 Java ME实例

下面的代码清单展示了在 Java ME[①]设备上获取位置的例子，比如在黑莓或三星、摩托罗拉、诺基亚或索爱的一些机型上。

代码清单 7-3 Java ME 上获取位置

```
Criteria cr= new Criteria();

cr.setHorizontalAccuracy(500);                         ❶ 设定选取位置的判断标准

LocationProvider lp= LocationProvider.getInstance(cr);

Location l = lp.getLocation(60);                       ❷ 获取提供者实例

Coordinates c = l.getQualifiedCoordinates();

if(c != null ) {                                       ❸ 请求位置

  double lat = c.getLatitude();                        ❹ 使用坐标信息
  double lon = c.getLongitude();

}
```

这段代码显示的是如何设置定位精度，这里水平精度是 500m❶，从系统中调出 LocationProvider 函数❷，获得位置值，设置 1 分钟的超时等待时间❸，然后使用它❹。

7.3.4 Palm webOS实例

Palm webOS 平台[②]上的服务可以访问低端硬件，比如 GPS 和加速度计数据，也可以访问高端数据服务，比如 Palm Synergy、云服务和任何其他网络服务 API。

下面的代码清单展示了在 webOS 设备（比如 Palm Pre）上获取位置的例子。

---

① mng.bz/o246

② http://developer.palm.com/

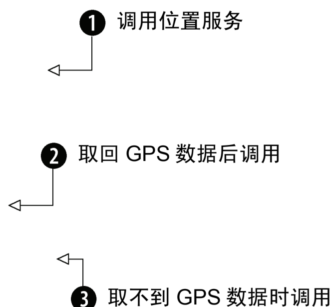
代码清单 7-4　在 webOS 上访问位置服务

```
this.controller.serviceRequest(
    ➥'palm://com.palm.location', {

  method:"getCurrentPosition",
  parameters: {},

  onSuccess: this.onSuccessHandler,

  onFailure: this.onFailureHandler
  }
);
```



这段代码显示的是如何发出位置请求❶，如何设置回调函数。调用的具体函数有赖于请求是成功❷还是失败❸。要注意的是如果把服务设计成为隔一段时间就返回一系列的结果，比如 GPS 跟踪数据，那么就会多次调用 `onSuccess` 函数。

7.4　小结

在设计 LBS 时要记住下面有关连接的指导原则：

- 妥善处理服务不可用的情况。经常有很多原因都可能造成用户位置不可用。
- 支持的所有定位方法都会有设备不适用的时候，比如在隧道中或在飞机上。
- 让用户决定信息发布与否。
- 设备支持不依赖位置供应商。
- 使用方法决定位置确定时间的长短。过长延时可能造成结果不能用，比如在导航应用中。要及时让用户了解情况。
- 一般网络辅助定位方法会快速累计位置服务费，因此不要过度使用收费服务。
- 减少电池使用。不需要时关掉 GPS。
- 根据具体情况决定用什么样的定位精度。如果对精度要求不高，就可以让基础服务提供低精度定位。
- 始终保护用户位置数据的安全。

第 8 章

服务器端集成

8

本章内容

- ❑ 创建高性能服务器
- ❑ 为服务器挑选数据库
- ❑ 高性能的秘诀
- ❑ 第三方 GIS 平台

移动应用的好坏取决于后台支撑它的服务器。连接移动应用的服务器向该应用提供所有共享的参考数据，处理移动应用发来的所有的用户请求，必要时还要更新后台数据库。服务器能够确保移动应用与最新的数据和信息同步，也能确保只有经过授权和许可的用户才能执行某些操作。服务器一般都放在一个地方，但云解决方案正变得越来越流行。特别是对一个有着大量兴趣点（POI）数据库和需要强大搜索功能的 LBS 应用来说，服务器更重要。如果你的服务器不能正确及时地响应用户需求，用户会马上从你的 LBS 应用切换到别人的应用上去。在这个竞争激烈、创业廉价的年代，不想尽办法用最佳方法执行后台服务，后果难以想象。

在这一章中，我们会了解 LBS 服务器要做什么，用一些代码实例介绍如何创建服务器，另外介绍优化整个基础设施的一些技巧和指标。首先，看一下在 LBS 应用中服务器做什么。

8.1 服务器的功能

一个 LBS 服务器能向移动客户端提供大部分的功能。LBS 服务器可能是运行一个简单数据库的单台服务器，也可能是囊括很多数据库的可扩展的云解决方案。无论服务器有多么地复杂或者简单，它们都有一些共同的特点，能提供一系列常见的功能。现在来看看典型 LBS 服务器的功能以了解其工作范畴和责任：

- ❑ 终端用户管理——终端用户的登录和注销，根据用户权限访问和更新数据。在多数应用中，用户必须登录才能访问特定的资源，例如，他们自己此前在应用中创建的内容。在基于位置的社交网络上，用户必须登录才能更新他们的位置和其他资料。
- ❑ 地图贴片服务——向客户提供地图贴片服务。LBS 服务器能向用户发送首选风格（地形、卫星、混合等）的适宜地图贴片。更多详情请参照 3.3 节和 8.6 节。

- 动态实体的位置和状态管理——插入、更新以及删除动态实体的位置，比如跟踪的用户和车辆。特别对基于位置的社交网络，每个用户都是一个动态实体，都可以在地图上移动。LBS 服务器必须跟踪系统中每一个动态实体的位置。
- 用户产生的内容管理——插入、更新以及删除由用户产生的内容（UGC），比如对企业或最喜爱的地方的评论。在大部分的 LBS 应用中，用户都可以创建位置内容，比如公园中最喜欢的地点或者在哪里举行生日宴会。LBS 服务器能够保存和索引所有的 UGC。
- 兴趣点（POI）管理——导入和管理多种来源的 POI（如图 8-1 所示）。LBS 服务器从第三方导入 POI 数据，对其进行索引，这样就能给客户供应相关的 POI 资料了。

图 8-1　LBS 应用中可见各种各样的 POI

- 动态搜索——围绕一个特定位置搜索和返回动态实体，比如用户。除了返回的是动态实体之外（如用户和车辆），这种搜索与静态搜索是一样的。
- 路线选择——查找和返回两个已知地点的路线。LBS 服务器能够查找两个已知点之间的最好路线，转而向移动客户反馈导航信息，“围栏”（fencing）如图 8-2 所示。
- 警报——对某些事发出警报，比如遇到障碍。当用户进入某区域（静态围栏）或两个动态实体互相接近（动态围栏）时，就能向客户发送警告。

现在你已经知道了服务器的各种功能，再让我们了解一下如何构造服务器，为各种各样的客户服务。虽然服务器与移动客户的交流方法有很多，但是最好是重用现有的软件构件使常用方法标准化。例如，你可以在服务器和移动应用之间创建一个专属的通信机制，不过这样一来，任何

开源的服务器或被广泛使用的客户程序库就不能使用了。另外，如果将来也想让其他人能与你的服务器通信，就会发现自己身处困境了，因为他们要学习你的专有通信方式才能与你的服务器交流。现在再看看服务器和移动应用之间最普遍的交流方式。

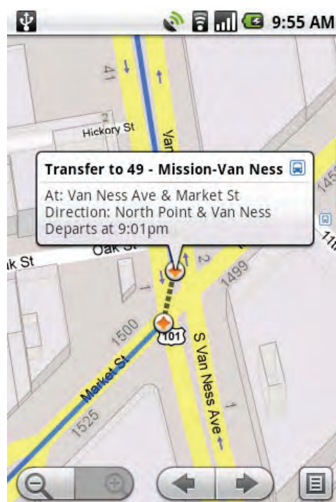


图 8-2　路线请求结果显示了详细的转向指示

8.2　服务器 API

服务器的功能通常通过调用 HTTP API 来实现，比如下面的例子：

http://api.example.com/poi/get/?lat=45&lng=–72

在这个例子中，调用的 API 会返回所有纬度 45 度、经度 72 度附近的 POI。构成 API 的方法叫做 RESTful。此 API 调用的结果有很多的格式，最常用的是 XML 和 JSON，它们是简单的文本格式，用一般的开源程序库都能很容易地解析。首先来看一看 RESTful API 调用。

8.2.1　REST

REST（Representational State Transfer，表述性状态转移）是一种分布式超媒体系统（如万维网）的软件构架风格。

REST 式的架构由客户端和服务器组成，如图 8-3 所示。客户端向服务器发出请求，服务器处理这些请求并返回适当的响应。请求和应答就构成了资源表现的转移。资源可以是任何连贯和有意义的概念。资源表现通常是一份截取资源当前状态或即将到达的状态的文件，比如代码清单 8-1 中的 `poi`。然后，资源 `poi` 就用于我们例子中 get 操作。

服务器通常会为了实现跨平台的兼容性而以 XML 或者 JSON 的形式返回一种文本格式的数据。

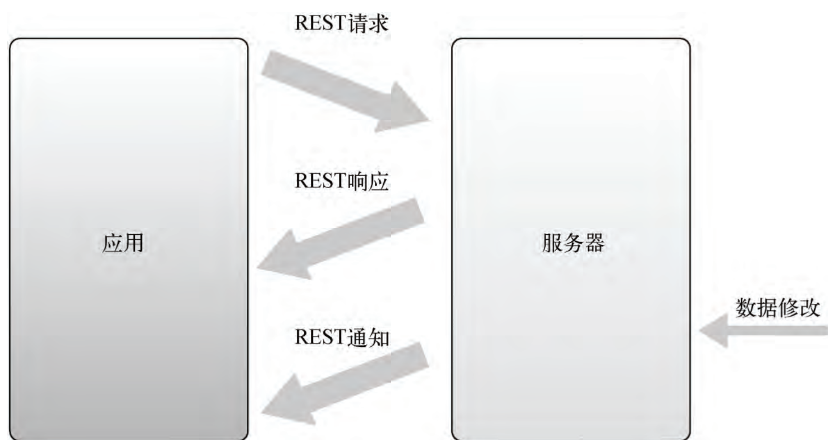


图 8-3　应用和服务器利用 REST 交换数据。应用向服务器发出 REST 请求，服务器向应用发出 REST 响应和通知

8.2.2　XML数据交换格式

XML（Extensible Markup Language，可扩展标记语言）是一种简单的基于文本的数据交换格式，是一套将文档编码为文本的规则。HTML 也是 XML 的一种形式，人们已开发了数百种基于 XML 的语言，包括 RSS、Atom、SOAP 以及 XHTML。XML 是一种常见的数据交换格式，因为用 XML 可以很容易地表现任何种类的结构化数据。下面的程序是包含了一些 POI 的简单 XML 实例。

代码清单 8-1　XML 实例

```
<?xml version="1.0" encoding='UTF-8'?>
<pois>
    <!-- This is an XML comment -->        <-- XML 代码中的注释

    <source url="http://example.com"/>     <-- ❶ 源 XML 对象包含属性 URL

    <poi>                                  <-- ❷ poi XML 对象包含其他 XML 对象
        <name>Grand Central Station</name> <-- ❸ 名称 XML 对象的值
        <rating>5.0</rating>
    </poi>
    <poi>
        <name>Empire State Building</name>
        <rating>5.0</rating>
    </poi>
</pois>
```

这个例子显示了 XML 对象可以包含属性❶、其他对象❷和数值❸。例如 poi 标签包含着名称和等级标签。

XML 是一种综合的文本格式，能够表示任何数据，但是要考虑服务器和客户端之间的传输

时间的话，它就可能太笨重了。正因为如此，尽管 XML 是一种常用的数据交换格式，但随着更轻量级 JSON 格式的出现，用 XML 的人就少多了。

8.2.3 JSON数据交换格式

JSON 是 JavaScript Object Notation 的简称，是一种轻量级的计算机数据交换格式。它是基于文本的、表现简单数据结构和联合数组（称作对象）的一种人类可读的格式。

LBS 应用程序要从服务器中检索大量的位置、地图和 POI 信息，当然需要注意传输时间。而采用重量级的数据格式会增加终端用户不必要的等待时间，这会毁了你的应用。

代码清单 8-2 是 JSON 格式的相同的数据实例。正如此例展示的那样，它包含了一些 POI。因为 JSON 相较于 XML 是较轻量级的格式，所以在客户端和服务器之间需要交换大量数据时，要优先考虑。

代码清单 8-2　一个 JSON 例子

```
      {
// This is a JSON comment                    <- JSON 的注释
          "pois" : {                         <- ❶ pois 键包含键值对
              "source": "http://example.com",   <- ❷ source 键的值是 URL
              "poi": [
                  {
                   "name": "Grand Central Station",
                   "rating": "5.0"
                   },
                   {
                    "name": "Empire State Building",
                    "rating": "5.0"
                   }
              ]
          }
      }
```

在这个例子中，一个密钥可以包含很多键值对❶和一个样本键值对❷。例如，poi 标签就有两个对象，名称和等级标签。可见，通过用轻量级 JSON 格式发送和接收数据，能为全部的 LBS 应用程序节省大量的时间。

现在让我们看一下怎样在服务器上存储和操作 LBS 数据。

8.3 空间数据库

空间数据是 LBS 应用的一个关键部分，也被称为地理空间数据或地理信息。空间数据能识别地物的地理位置和边界，比如自然或人造特征、海洋或其他地貌。空间数据通常存储为坐标和拓扑结构，是能在地图上绘制的数据。LBS 服务器需要有效地存储和访问空间数据，因为这是其主要功能。

对 LBS 服务器来说，能够有效存储和操控空间数据的数据库是非常重要的。因为每一个数

据结构和所有的 LBS 功能都需要进行空间操作，比如存储对象的经纬度，或者要进行关于位置数据的操作，比如寻找附近的对象。

空间数据库通常支持图 8-4 所示的数据类型。正如你所见到的，存储、检索和操控这些空间数据类型是密集型计算，需要一个高效的数据库来处理。现在我们开始看一下在你的 LBS 服务器中可以使用的空间数据库。

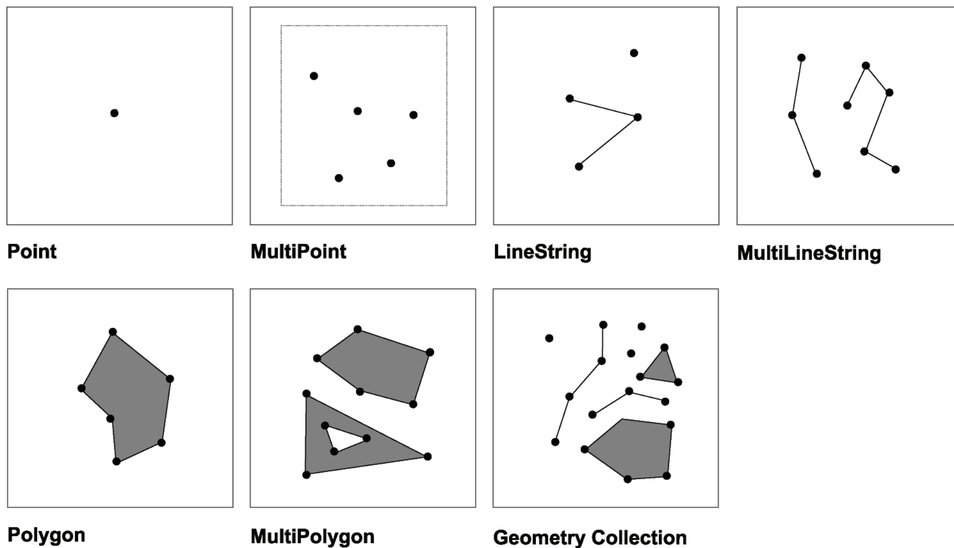


图 8-4　常用的空间数据类型

8.3.1　PostgreSQL和PostGIS

PostgreSQL 和 PostGIS 是 LBS 应用中常用的空间数据库。PostGIS 能提供空前的传输速度和功能，这对 LBS 应用来说是非常重要的。PostgreSQL[①]是一种对象关系数据库管理系统（ORDBMS）。它的发布遵循 BSD（Berkeley Software Distribution，伯克利软件套件）式许可证，因此是自由的开放源软件。

PostGIS[②]是一种开源软件程序，能支持 PostgreSQL 的地理对象。实际上，PostGIS 空间允许 PostgreSQL 服务器的访问，并可以将其作为空间数据库的后台。PostGIS 遵守 Open Geospatial Consortium[③]（开放地理空间联盟）的 “Simple Features for SQL” 规范[④]。同样地，PostGIS 包括下面的功能：

① http://www.postgresql.org/
② http://postgis.refractions.net/
③ http://www.opengeospatial.org/
④ http://www.opengis.org/docs/99-049.pdf

- Points、LineStrings、Polygons、MultiPoints、MultiLineStrings、MultiPolygons 以及 Geometry Collections 等几何类型，如前所示
- 确定几何体之间相互作用的空间判断
- 确定地理空间尺寸（如面积、距离、长度和周长）的空间运算符
- 确定地理空间几何操作（如并集、差分、对称差分和缓冲）的空间运算符
- 快速空间检索的 R-tree-over-GiST（Generalized Search Tree，通用搜索树）空间索引
- 选择性索引支持，为混合空间/非空间检索提供高效的查询计划

代码清单 8-3 显示的是各种不同类别几何体类型的例子。

代码清单 8-3　支持 PostGIS 的几何类型实例

```
Point
Example: POINT (10 10)
LineString
Example: LINESTRING( 10 10, 20 20, 30 40)
Polygon
Example: POLYGON ((10 10, 10 20, 20 20, 20 15, 10 10))
Multipoint
Example: MULTIPOINT(10 10, 20 20)
Multipolygon
Example: MULTIPOLYGON(((10 10, 10 20, 20 20, 20 15, 10 10)),((60 60, 70
70, 80 60, 60 60)))
GeomCollection
Example: GEOMETRYCOLLECTION(POINT (10 10), POINT(30 30), LINESTRING(15
15, 20 20))
```

这段代码展示了如何在 PostGIS 中声明各种类型的几何体，比如 Point、LineString、Polygon、MultiPoint、MultiPolygon 和 GeometryCollection。例如，在 LBS 应用程序中住宅区边界能通过一个多边形或多个多边形有效地表现出来。

PostGIS 基于优化的轻量级几何图形和索引，减少磁盘和内存占用。利用轻量级的几何图形有助于服务器增加从物理磁盘存储器迁移到 RAM 的数据数量，从根本上提高查询性能。现在让我们看一看另一种流行的数据库 MySQL，它具有一些额外的空间支持功能。

8.3.2　MySQL空间支持

MySQL（现在属于 Oracle 公司）是一种关系数据库管理系统（RDBMS），全球安装超过 6 百万台。MySQL 经常用在需要功能全面的数据库管理系统的软件项目中。

MySQL 支持空间扩展功能，允许创建、存储和分析地理要素。空间索引也能优化搜索操作。借助于早前设计的各种多维索引方法，MySQL 就能优化空间搜索功能。最典型的空间搜索如下：

- 为所有包含给定点的对象提供 Point 查询
- 为所有给定重叠地区的对象提供 Region 查询

MySQL 空间扩展支持以下几何类：

- Geometry（不可实例化）

- ❑ Point（可实例化）
- ❑ Gurve（不可实例化）
- ❑ LineString（可实例化）
- ❑ Line
- ❑ LinearRing
- ❑ Surface（不可实例化）
- ❑ Polygon（可实例化）
- ❑ GeometryCollection（可实例化）
- ❑ MultiPoint（可实例化）
- ❑ MultiCurve（不可实例化）
- ❑ MultiLineString（可实例化）
- ❑ MultiSurface（不可实例化）
- ❑ MultiPolygon（可实例化）

例如,LBS 应用程序中的一条公交车路线可以通过 MultiLineString 表示,这个能利用 MySQL 有效地处理。尽管如此，MySQL 的空间支持并不如 PostGIS 的那样高效和完善。现在让我们看看 Microsoft 数据库中的空间支持。

8.3.3　Microsoft SQL Server空间支持

Microsoft SQL Server 是由微软研制的一种关系模型数据库服务器。运行在 Microsoft 平台上的 LBS 应用需要了解 Microsoft SQL Server，看这种服务器是否符合需求。

SQL Server 2008 在 SQL Server 产品组件上添加了地理空间支持功能。这样就能在 SQL 表中（以点、线和多边形的形式）存储空间数据，用一系列的函数操控这些数据了。同样它也包括了能够支持这些函数执行的新空间索引。

SQL Server 2008 支持两种不同的空间数据类型：

- ❑ GEOMETRY——这种数据类型存储投影平面表面数据。
- ❑ GEOGRAPHY——这种数据类型存储椭球面模型数据。

几何类型包括如下：

- ❑ Point——点是代表一个位置的对象。它常有一个 X 坐标和一个 Y 坐标，此外可能有一个海拔 Z，一个尺寸 M。
- ❑ MultiPoint——MultiPoint 对象是很多点的集合。它不同于 LineString 和 Polygon，因为这个集合中的点没有隐含的联系。因此，一个 MultiPoint 对象没有边界。
- ❑ LineString——LineString 又是一个点的集合。它不同于 MultiPoint 对象，因为其中的点是按顺序排列的，LineString 对象也能表示连接各点的线段。
- ❑ MultiLineString——MultiLineString 是 LineString 的集合。
- ❑ Polygon——Polygon 是表示二维表面上的点集合。一个 Polygon 可能包含一个外部环和若

干内部环。对一个有效的 Polygon 对象来说，其内部环不能相互交叉。

- MultiPolygon——MultiPolygon 是 Polygon 的集合。
- GeometryCollection——GeometryCollection 是所有几何体（或地理）对象的集合。

下面的列表展示了如何创建空间表以及如何将数据导入到表中。首先创建带几何字段的表格。然后在表格中插入数据。

代码清单 8-4　创建空间表和导入数据的代码示例

```
CREATE TABLE Districts ( DistrictId int IDENTITY (1,1), DistrictName
nvarchar(20), DistrictGeo geometry); GO        ❶ 在数据库中创建新表
CREATE TABLE Streets ( StreetId int IDENTITY (1,1), StreetName
nvarchar(20), StreetGeo geometry); GO
INSERT INTO Districts (DistrictName, DistrictGeo) VALUES ('Downtown',
geometry::STGeomFromText ('POLYGON ((0 0, 150 0, 150 150, 0 150, 0 0))',
0));                                           ❷ 在表中插入值
INSERT INTO Districts (DistrictName, DistrictGeo) VALUES ('Green Park',
geometry::STGeomFromText ('POLYGON ((300 0, 150 0, 150 150, 300 150,
300 0))', 0));
INSERT INTO Districts (DistrictName, DistrictGeo) VALUES ('Harborside',
geometry::STGeomFromText ('POLYGON ((150 0, 300 0, 300 300, 150 300,
150 0))', 0));
INSERT INTO Streets (StreetName, StreetGeo) VALUES ('First Avenue',
geometry::STGeomFromText ('LINESTRING (100 100, 20 180, 180 180)', 0)) GO
INSERT INTO Streets (StreetName, StreetGeo) VALUES ('Mercator Street',
geometry::STGeomFromText ('LINESTRING (300 300, 300 150, 50 51)', 0)) GO
```

在这个例子中，可以看到如何创建新的表❶，在表中插入值（行）❷。我们创建了 Districts 和 Streets 表。然后在 Districts 中插入三行，在 Streets 中插入两行。

你已经看到了 Microsoft 所提供的功能，再了解下世界最大数据库公司提供的 Oracle 所具有的功能。

8.3.4　Oracle空间

Oracle Spatial[①]构成了一个经 Oracle 数据库个别许可的可选组件。如果你需要一个企业级的 LBS 数据库或者在现有的 Oracle 数据库上建立一个 LBS 数据库，那么 Oracle Spatial 是 LBS 应用程序可用的另一种方法。Oracle Spatial 帮助用户管理 Oracle 数据库中原生类型的地理和位置数据，也有可能支持各种不同的应用，从自动绘图/设备管理和地理信息系统，到无线位置服务和基于位置的电子商务。

Oracle Spatial 包含的功能如下：

- 规定了存储方式、语法和支持几何数据类型语义的模式（schema）
- 空间索引系统
- 执行感兴趣面查询，空间连接查询以及其他空间分析运算的运算符、函数和程序

① http://www.oracle.com/technology/products/spatial/index.html

- 实用工具和调优操作的函数和程序
- 用于操作拓扑中的节点、边界和面数的拓扑数据模型
- 在网络中表现能力或对象（节点和链接模型）的网络数据模型
- 存储、索引、查询、分析以及交付栅格数据（光栅图像和格网数据及其相关联的元数据）的 GeoRaster（栅格）特性

就像 Oracle 一样，IBM 提供的 DB2 数据库也是工业级的解决方案。

8.3.5　IBM DB2 Spatial Extender

DB2 Spatial Extender（IBM DB2 空间扩展器）能够将与传统空间数据（地物的位置信息）的文本和数字数据一起通过 DB2 Universal Database 存储、管理和分析。

有了这个功能，你就能生成、分析和利用有关地物的空间信息，比如办公大楼的位置或洪水区域的大小。DB2 Spatial Extender 扩充了 DB2 Universal Database 的功能。它采用一套先进的空间数据类型来表现点、线和多边形这样的几何图形，提供了利用这些新数据类型进行交互操作的很多功能和特性。这样就能整合空间信息和商业数据，使你的数据库变得更智能。

现在你已经了解了所有的数据库和 API 选项，接下来让我们看看如何让你的服务器在用户数量很庞大的情况下保持最佳的性能。

8.4　性能

对很多耗时的操作，比如搜索和更新大量的实体来说，充分优化服务器性能势在必行。正如你能预料的那样，随着数据量的增加，完成复杂运算所需的时间呈指数性增长，因此采用最优化技术实现一个响应灵敏的服务器是很重要的。缓存是其中一种能增强服务器性能的最有效的技术。

缓存

LBS 服务器经常会遇到类似的查询命令，通过访问数据库反复地读取相同的对象。缓存查询结果以及在内存中缓存对象能节省大量的时间。在某些情况下，90%的改进工作是有关非缓存服务器的。

最常用的内存对象缓存机制是 Memcached（分布式缓存）。Memcached[①]是由 Danga Interactive 为 LiveJournal 开发的一种多用途的分布式内存缓存系统，但是现在很多其他的网站也在使用。它常被用来加速由数据库驱动的动态网站，通过缓冲内存中的数据和对象来减少必须读取的外部数据源（比如一个数据库或 API）次数。分布式缓存由开放的自由软件许可证授权释出。

Memcached 使用方便，如图 8-5 所示。你能在 Memcached 中缓存先前查询的结果，而不用对每一个请求都做一次数据库动态查询，只要可能就使用缓存值从而避免增加数据库负担。每一

① http://memcached.org/

次查询数据库的时间会比使用缓存数据多耗费几个数量级的时长。

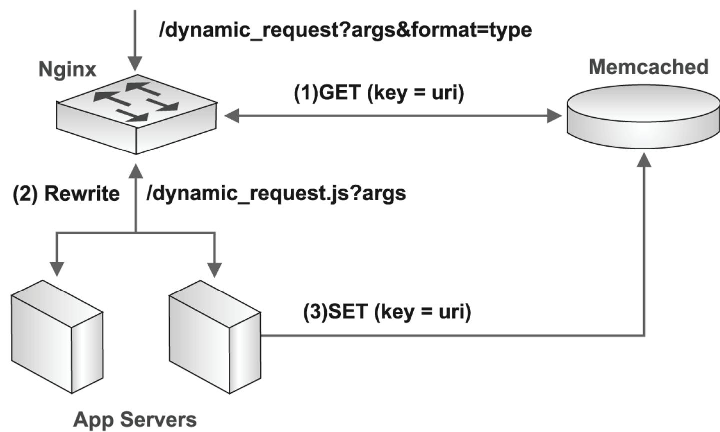


图 8-5 Memcached 的基本运算（来源：mng.bz/001q）

下面的例子中的伪代码访问数据库以读取用户对象的内容：

```
function get_user_from_id(int id) {

  result = db_select("SELECT * FROM user WHERE id = ?", id);
  return result;

}
```

这段代码每次都要直接与数据库相互作用，因此会运行得很慢。

下面这些列表展示了在使用 Memcached 之后，缓存用户对象的相同代码，因此调用此函数时，就会从内存中返回用户对象，而不是读取数据库。

8

代码清单 8-5 Memcached 代码示例

```
function get_user_from_id (int id) {

                                                        ❶ 检查对象是否存在
     result = memcached_fetch("userindex:" + id);

     if (!result) {                                     从数据库中取得 ❷

        result = db_select("SELECT * FROM user WHERE id = ?", id);

        // 把对象插入缓存
        // 下次就可从中找到
         memcached_add("userindex:" + id,  result);
     }
     return result;
 }
```

这段代码在查询数据库❶之前检查了 Memcached，只要用户对象不在 Memcached 之中，就从数据库中提取用户对象❷。正因为这样，与代码清单 8-6 中的代码相比，这段代码运行得要快很多。

现在你已经知道了一些有关创建高性能 LBS 数据库的问题，让我们看个例子。

8.5　返回 POI 示例

现在看一下如何写一个简单的 LBS 服务器请求，这个请求能够返回指定位置附近的 POI。

首先要创建一个数据库表，导入我们的数据，如下所示。这段代码首先创建了一个表，然后在表格中插入数据。

代码清单 8-6　创建一个 PostGIS 表，在表格中导入 POI

```
create table poi_locations (gid int4, poi_name varchar);          ❶ 创建表
select AddGeometryColumn
('db_mapbender','poi_locations','the_geom','4326','POINT',2);
insert into poi_locations values ('1','Times Square',GeometryFromText
('POINT(-72.060316 45.432044)', 4326));                  ❷ 在表中插入记录
insert into poi_locations values ('2', 'Empire State',GeometryFromText
('POINT(-72.6764 45.8916)', 4326));

...
```

这段代码创建了一个表❶，插入很多代表 POI 的行❷。

现在让我们看看如何访问这些表，以及如何在表中利用 PHP 搜索数据，如代码清单 8-7 所示。PHP 是最常用的服务器语言，在开发者社区和开源程序库方面有大量的支持。Ruby、Python 和 Java 是其他流行的服务器端语言。

代码清单 8-7　用 PHP 代码搜索数据库中附近的 POI

```
<?php

// get.php                 ← 请求 URL http://x.com/poi/get/?lat=45&lng=-72

$lat = $_REQUEST['lat'];
$lng = $_REQUEST['lng'];          ❶ 取得经、纬度

$dbconn = pg_connect("dbname=myapp");          ❷ 连接数据库

$query = "SELECT the_geom, poi_name FROM poi_locations WHERE
            ST_DWithin(the_geom, 'POINT($lat $lng)',1000)";     ❸ 查询数据库

$result=pg_query($dbconn, $query);

while ($row = pg_fetch_row($result)) {          ❹ 循环结果

    echo json_encode($row);          ❺ 返回 JSON
}
```

```
pg_close($dbconn);          <-- ❻ 关闭连接

?>
```

这段代码首先找到请求的 lat 和 long❶，然后连接数据库❷，获得所有附近 POI 的 lat/long❸。以 JSON❺格式返回每一个 POI❹，然后断开连接❻。这是大多数 LBS 服务器运作常见的例子。如果你继续运行该代码，LBS 服务器就会不断给你发送更新的 POI 信息。

有时候，你也许能利用第三方的 LBS 服务器，而不用从头做起。现在看看有哪些第三方 LBS 服务器。

8.6　第三方 LBS 服务器

有很多 LBS 服务器可以使用或能得到许可使用。如果你的应用不需要定制的服务，就可以考虑使用这些 GIS 系统作为后台。其中有些 GIS 系统甚至可以让你通过内置的脚本语言做较小的自定义。

8.6.1　MapServer

MapServer[①]，如图 8-6 所示，是一个搭建空间网络应用的开源研发环境。它是由 University of Minnesota（明尼苏达大学）开发的。最初由 NASA 支持研发，NASA 需要一种途径使其卫星影像可以为公众所用。现在 MapServer 是 OSGeo[②]的一个项目，世界各地越来越多开发人员参与维护。

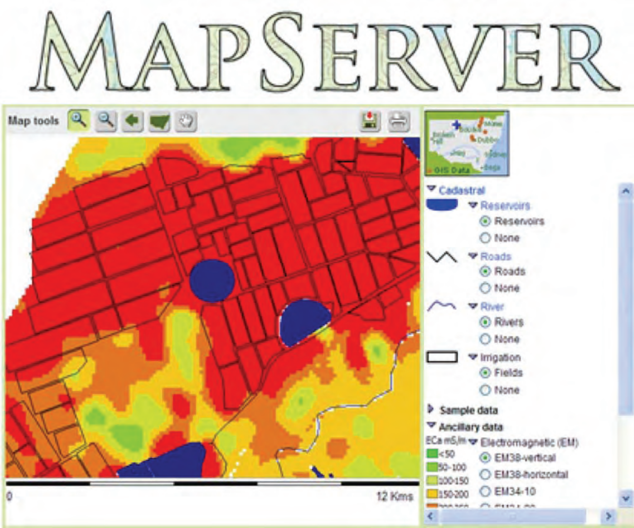


图 8-6　MapServer 客户端视图

① http://www.mapserver.org/
② http://www.osgeo.org/

MapServer 不是一个功能十分完善的 GIS 服务器，但是它已经发展了很长的时间，很多应用都可以方便地使用。它支持先进的地图输出，具有以下特点：

- 按比例的特征图和应用执行
- 包括标签冲突调节的特征标签
- 可定制的、模板驱动的输出
- TrueType 字体
- 地图元素自动化（比例尺、参考地图和图例）
- 使用条理分明的或基于正则表达式的基本类的专题地图
- 支持流行的脚本语言和开发环境（PHP、Python、Perl、Ruby、Java 和.NET）
- 跨平台支持（Linux、Windows、Mac OS X、Solaris 及更多）
- 支持许多的 Open Geospatial Consortium (OGC)标准，比如 WMS（客户端/服务器），非事务型 WFS（客户端/服务器）、WMC、WCS、Filter Encoding、SLD、GML、SOS 以及 OM

MapServer 的优势在于其稳定性和来自开源社区的广泛支持。如果你需要更可靠的商业支持，那么 ESRI 的 ArcGIS 服务器是一个很好的选择。

8.6.2　ESRI ArcGIS Server

ArcGIS Server[①]是由 ESRI 开发的提供适合网络空间数据服务的 GIS 软件包。自 ArcGIS Server 9.2 版本以后，它也包括了空间数据管理软件（以前叫作 ArcSDE）。如果你正在为不断扩展的服务器寻找专业支持和帮助，那么 ArcGIS 是一个不错的选择。

ArcGIS 服务器支持.NET Framework 和 Java 编程语言开发。ArcGIS Server 能为浏览器、移动设备和台式机提供数据。它支持互操作性标准，比如 OGC 和 W3C。地图服务、地理编码服务、地理数据管理服务、地理处理服务、虚拟的全球服务以及网络分析服务都能通过 SOAP API 和 REST API 来访问。

现在让我们看看另一个商业 LBS 服务器，Maptitude。

8.6.3　Maptitude

Maptitude[②]是由 Caliper Corporation 开发的绘图软件程序，用户可以利用该程序查看、编辑和整合地图。这个软件和相关技术是为了分析内置数据或者自定义的外部数据，以及地理可视化而设计的。其 Microsoft Windows 平台的商业应用包括下面的功能：

- 创建地图显示
- 增强地图报告和展示
- 找到在数据库表和电子表格中看不到的地理模式
- 回应影响经营活动的地理问题

---

① http://www.esri.com/software/arcgis/

② http://www.caliper.com/maptovu.htm

❑ 与工作组、部门或组织分享地理数据

Maptitude 的主要目标是商业用户，但它与很多部门的各级 GIS 市场都有竞争。它与 Microsoft Office 整合，能使用包括 Microsoft Excel 在内的不同数据源，提供了专有的类似 BASIC 的编程语言（Caliper Script），还有一个开发界面（GISDK），支持 Maptitude 的自动操作。

8.6.4　GeoMedia

GeoMedia[①]是 Intergraph Corporation 的 GIS 软件产品系列中的一套软件。它是专门为 Microsoft Windows 环境开发的可作为客户端或服务器的软件，如图 8-7 所示。

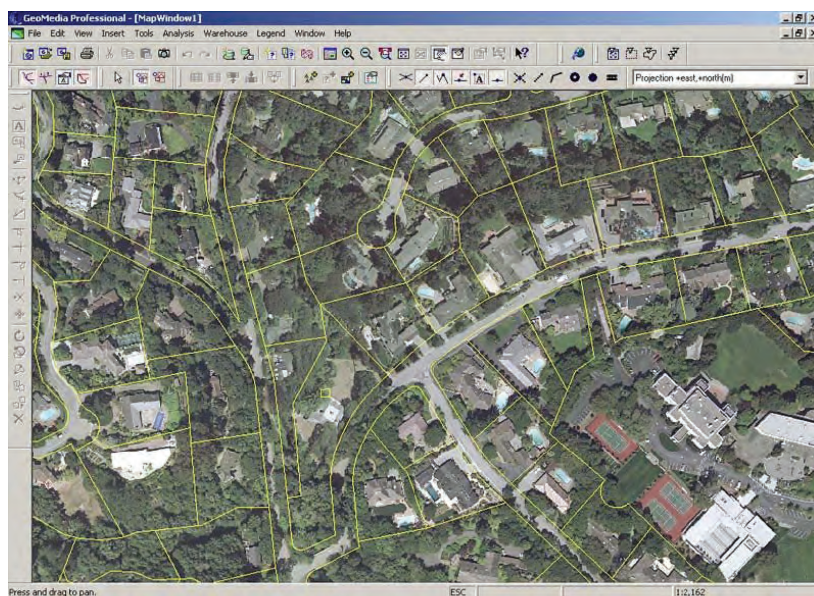


图 8-7　GeoMedia 客户端视图

GeoMedia 的核心技术使得它能同时从许多的 GIS 数据源中直接读取数据，这些数据源包括

❑ Shapefiles (ESRI)

❑ ESRI Coverage (ESRI)

❑ ESRI SDE via FME plug-in for GeoMedia (ESRI)

❑ AutoCAD DWG、AutoCAD DXF (Autodesk)

❑ MicroStation DGN (Bentley Systems)

❑ Oracle Spatial (Oracle Corporation))

❑ MapInfo (MapInfo)

❑ FRAMME (Intergraph)

① mng.bz/B8w8

- ❑ Modular GIS Environment （MGE）（Intergraph）
- ❑ 基于 Microsoft Access 或 Microsoft SQL Server 的 GeoMedia 仓库（Intergraph）

接下来，了解另一种商业产品，MapInfo Professional。

8.6.5　MapInfo Professional

MapInfo Professional①是由 MapInfo Corporation 研发的一个桌面制图系统软件产品。MapInfo Professional 能够在一张地图上合并和显示各种来源、不同格式和项目的数据。这套软件能够在同一地图上叠加光栅和矢量图层；前者可以半透明显示，这样它们的用途就不只是背景布幕了。MapInfo 通常用于分析预制的地图数据图层。

接下来了解一下来自 Microsoft 的非常先进的 LBS 服务器，MapPoint 服务器。

8.6.6　Microsoft MapPoint

MapPoint Web Service②是一种基于 XML 的网络服务，它能使开发人员将地图、行车路线以及邻近搜索这样的 LBS 服务与其应用和业务流程整合起来，如图 8-8 所示。

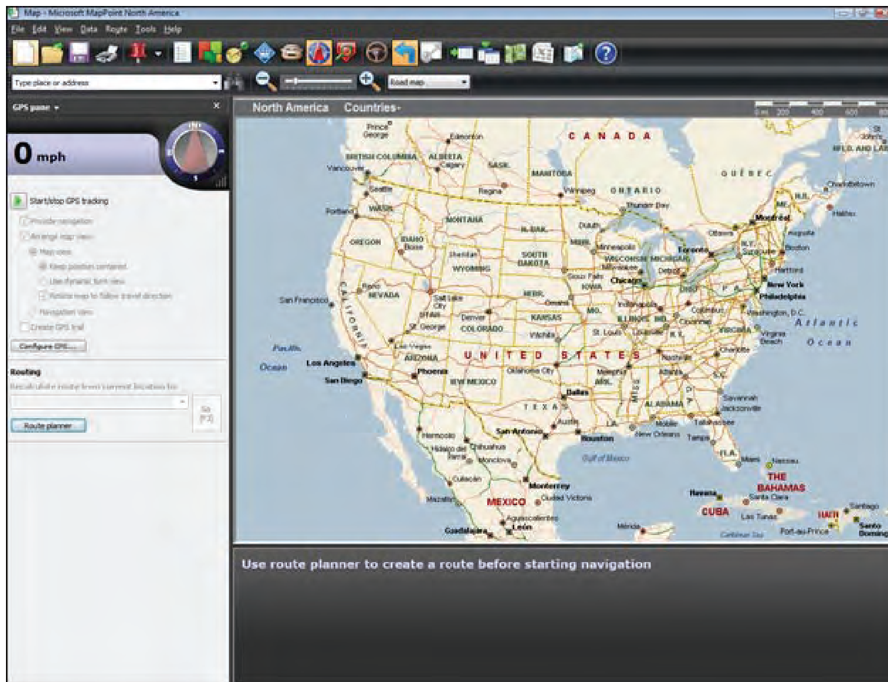


图 8-8　Microsoft MapPoint 客户端视图

① mng.bz/3Ii3
② http://www.microsoft.com/mappoint/

MapPoint Location Server 是 MapPoint Web Service 的一个组成部分，它能将实时位置信息整合到商务应用中，能提供访问位置供应商的接口，使之与移动运营网络相结合。MapPoint Location Server 充当了这些网络和 MapPoint Web Service 的代理，将它们提供的实时位置数据和服务联系起来。

MapPoint Web Service 用户只要有许可协议，就可以使用 MapPoint Location Server。如果你正在开发一个企业级的 LBS 应用，需要可扩展和有力的支持，并有足够的预算，那么 MapPoint 将是不错的选择。

现在我们已经分析完所有可用的 LBS 服务器。仔细挑选一种服务器是非常重要的，因为它将涉及时间和金钱的大量投资。

8.7 小结

服务器是移动 LBS 应用中很重要的一个部分。如果挑选的方法、技术和数据库不恰当，你开发的移动应用程序就可能因为运行缓慢和反应迟钝而导致用户更换其他应用。

性能是非常重要的。数据库的规模增加了，其运行性能就会呈指数级下降，因此需要缓存所有能够缓存的内容。然后开始分析和研究服务器的请求，找到瓶颈，缓存更多的内容。在很多时候，分析服务请求能发现意想不到的问题。另外，需要在不理想的网络连接状态下，在所有目标平台上进行性能测试。也可以对服务器进行压力测试，设想应用大获成功会发生什么情况。

首先寻找第三方 GIS 系统解决方案。如果这些都不适合你的应用，就要定制自己的服务器。但是在很多时候，如果你所做的不只是获取和浏览数据，那么多数的应用都需要自定义的服务器。

无论是数据处理还是数据访问，要使你的服务器保持负载平衡。建立一个数据库备份，以便有事故发生时能继续提供服务。

下一章我们将会讨论与 LBS 应用相关的隐私问题。

Part 3

第三部分

LBS 应用商业化

在本书的最后部分，我们将会探索 LBS 移动应用程序开发的。商业化会让你努力开发的成果被人们广泛接受，让开发出的软件更成功更有利可图。

在本书前言中我们曾提过，有越来越多的应用开发人员也成为了创业者。

正因如此，在本书的最后，除了介绍如何使你开发的应用脱颖而出，也会介绍在企业创建、融资管理方面的经验。

与多数 Web 或移动服务一样，赢利模式是初创企业成功的关键。第 9 章将会讨论如何实现赢利，并向你展示如何为应用或服务配备正确的赢利模型。另外还要探索经久不衰的货币数字化服务的免费增值模型。

隐私保护仍是最重要的问题，所以第 10 章会介绍减小隐私恐慌的手段。

在第 11 章中，我们会提出一些有用的技巧，确保你开发的应用能在竞争日益激烈的应用商店中脱颖而出。最后一章将会介绍一些推荐的商业惯例，以保护你的想法和应用交易——这是非常重要，但又容易被忽略的部分。

读完本书的最后一部分，你就能够策划出一个胜券在握的商业策略，从而使你的应用在未来的移动应用程序世界中获得最大的成功机会。

第 9 章

LBS 商业化

9

本章内容

- ❑ 消费者是利润的来源
- ❑ 企业是利润的来源
- ❑ 货币化 IP

天下没有免费的午餐。

——佚名

由 Milton Friedman（米尔顿·弗里德曼）推而广之的这句古老格言，同样适用于移动和 Web 服务，而 LBS 也不例外。虽然 LBS 免费服务或补贴服务常用于促进早期的采用者在 LBS（尤其对基于位置的社交网络而言）上的尝试，但是 2008 年次贷危机造成的全球经济衰退使 LBS 必须面对创收的现实。

今天，移动初创企业的投资者渴望看到清晰、可靠、可持续的创收计划。仅为广大用户提供免费服务就足够融到资的日子已经一去不复返了。那时的企业家根本不会担心自己的公司赚不赚钱。只要吸引到有能力购买他们产品的大公司，他们就可以高枕无忧了。很快到了今天，LBS 每年所产生的总利润超过了 30 亿美元[①]。这意味着现在将想法变为有形的利润并不是可望而不可及的事了。

企业家们可选的盈利模式各不相同，值得注意的是有些模式是有周期性的，而且移动生态系统的发展周期越来越与实体经济的发展同步了。举个移动广告的例子。次贷危机之前，LBS 初创企业仅靠广告就能生存，但在 2008 年左右，仅靠广告就无法维持了。由于手机广告投放严重短缺，造成应用过剩，与此同时手机广告率直线下降。

随着竞争越发激烈，仅靠打广告维持发展的可能性非常小。虽然基于位置的广告已遍及全球，但距离发展成熟还早得很。这就造成特定领域内基于位置的广告供给和广告需求之间的不平衡，因此获得可持续性收入变得更为困难。尽管如此，那些有创意的企业家即使在困难时期依然收获颇丰。比如芝加哥的 Groupon。Groupon 创建于 2008 年年初，它能为本地交易（主要表现为基于

① “Location Based Services on Mobile Phones,” The Economist, 2010 年 3 月 4 日; 在 mng.bz/LT86 上能看到。

位置广告的另外一种更成熟的版本）提供很大的折扣。该公司 6 个月之后开始盈利，并且到了 2010 年，已产生了接近 5 亿美元的收入。

类似地，多数的初创企业也争先恐后地争取用户，好把他们变成付费用户，因此也遭遇到不少用户的抵触。这就迫使他们自我反省，在某些情况下，这些公司必须改变关注点，从企业对消费者（B2C）形式转变为企业对企业（B2B）的形式。

那么在革新 LBS 应用基础上，怎样才能建立一个稳固的业务模式呢？怎样才能从你花费了数小时、数周或者数月开发的应用中挣到钱呢?当然，这不仅取决于锁定哪些目标客户以及传递了多少感知价值，还要取决于当时特定的市场以及经济环境。关键就是要了解可供选择的范围，这样才能知道哪些对策能最好地适应当前的环境。对成功的初创企业来说，灵活性是其制胜的法宝（一次没有成功，就再努力尝试！）。

本章我们将会探索利用 B2C 模型和 B2B 模型创造利润的方法。对 B2C 模型，我们将付费访问与付费展示的不同之处。对 B2B 模型，我们将讨论针对手机“版面”（应用中的屏幕显示空间）收费与对应用获取的位置数据收费。

9.1 消费者是收入的来源

对 LBS 公司来说，直接向终端用户收费有很多明显的优势（至少从理论上是这样）——它更快速、更能深入大众市场，并且相较于 B2B 模式来说，障碍也更少。不利的方面就是消费者变得更善变、更不易琢磨（就像是对时尚的反应），并且如果经济环境不好，他们就会减少随意开支。

现在，在某些特定的情况下，很多新 LBS 初创企业会直接面向消费者。更常用的赚钱方法就是向终端用户提供服务，这种方式可以称作计费网关（gatewag charging），接下来我们将会介绍。

9.1.1 计费网关

计费网关涉及向访问一项服务或应用的终端用户进行收费。这些收费方式可以是一次性的，也可以是重复性的（一月一次或一年一次）。

一次性收费通常适用于用户从应用程序商店下载 LBS 应用，应用商店中含有内置的交易机制可以从用户账户中扣除费用（第 11 章会有更多的应用程序商店介绍）。一旦用户在手机上下载安装了该应用，那么对于用户使用该应用就不会进一步地收费了。在某些情况下，未来的软件更新需要支付额外的费用。

很明显，对应用收费非常诱人，但也要小心行事。靠应用收费就意味着相较于免费而言，应用下载量会大幅降低。在某些情况下，这个比例会达到 400：1（免费下载比收费下载）[①]。

图 9-1 是一个示例图，引进收费会导致下载量戏剧性加速，一个很典型的案例就是 Galaxy

① Bo Wang, Bokan Technologies, TechCrunch, March 22, 2009; mng.bz/YqZB.

Impact iPhone 应用。虽然该应用并不完全是 LBS，但是从一开始就提供收费应用，与将免费应用变成收费应用相比，结果并不一样。毫无疑问，免费的应用会有较高的下载量。

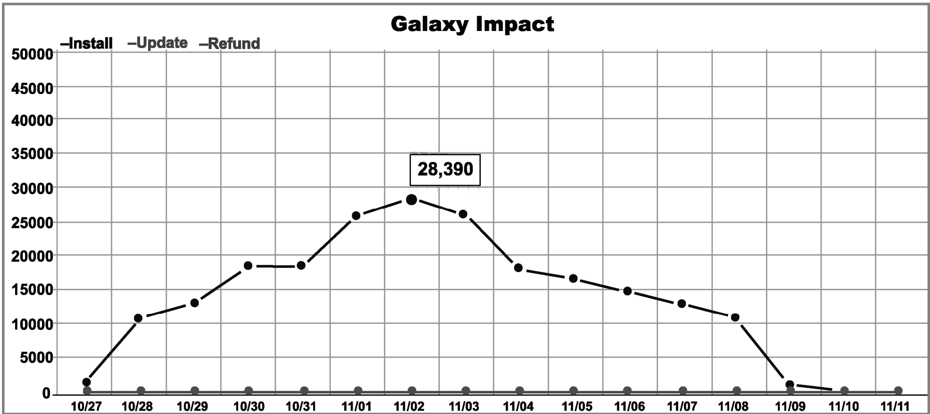


图 9-1　引进收费（在 11 月 9 日）对 Galaxy Impact iPhone 应用下载量的影响（来源：见上一页脚注）

对应用进行收费还必须权衡一次性收费和通过广告间接收费（本章后面会讨论）的关系。

一次性收费机制有一种变体，即先提供一个免费的基本应用（或“精简”版本），然后说服用户更新到收费的版本（通常提供额外的功能，更好的连接速度或者额外的内容）。这种模式称作自由增值，但也会涉及程序内购买，9.1.2 节将讨论到。值得注意的是，手机运营商的研究表明，在一个为期至少 1 年的时间段内，至少有 15%的消费者会从免费应用升级到收费服务[①]。这就意味着自由增值模式既可以将应用推广给更多公众，又可以在中期向用户收费。

对移动初创企业来说，重复性收费模式更具吸引力，因为这种收费方式能够保证一些连续的现金流，能促进其服务的“粘性”，订阅了一种移动服务的消费者不太可能停止订阅（尤其在订阅费用是通过移动运营商账单支付的情况下）。

像美国的 Loopt（在第 5 章中介绍过的）这样的服务，最初是通过移动运营商按月收取订阅费用的。现在，这种模式已被位置感知和基于位置的应用程序开发商们摒弃了（虽然一些导航辅助公司比如 Telmap 仍在使用）。

通常，对一些有着较高用户感知价值的服务来说，重复性收费会收到良好的效果。而对于休闲类的应用，比如基于位置的社交网络，采用刺激购买、一次性收费效果更好。

除了决定一次性收费还是重复性收费之外，开发者还需要考虑应用是否具有大众市场吸引力，是否有感知价值。这有助于决定应用的价格策略（免费或收费），也能帮助他们决定是否采用广告。

① Olivier Laury, Bouygues Telecom, “Finding the Right Strategies for Location-Based Services,” Seminar, Mobile World Congress, 2009 年 12 月。

图 9-2 以一个决策矩阵的方式提供了一个简单的模型，从而辅助开发者对消费者应用做出如下选择：

- 一个免费的精简版（缩减功能的基础应用），无广告
- 一个有广告支撑的免费应用
- 一个没有广告的收费应用
- 一个有广告的收费应用

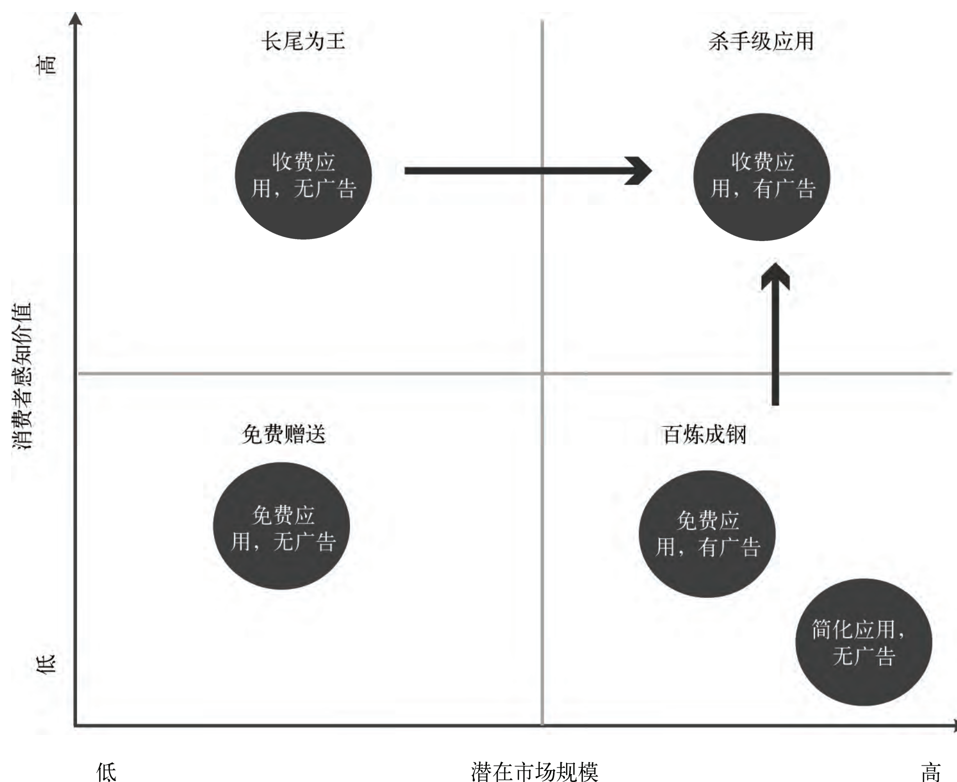


图9-2　应用价值/市场决策矩阵展示了根据新应用的感知价值和潜在市场大小制订的不同收费策略

“免费赠送”的应用显示在矩阵左下方。很明显，该类应用不可能有任何前途，尤其在没有广告支撑的情况下。但是对于低价值/低市场潜力的应用，这是唯一可能的选择了。这种情况常被大品牌的公司用于市场推广。也适用于自由增值模式，即首先引进一种基础版的免费应用，将来再增加高级功能。

“长尾为王”是市场中那些用户数量较少但是具有较高价值定位（在第 5 章中介绍过长尾的概念）的应用。开发者可以对应用收取较高的费用，因此不需要广告投入（这也能确保提高用户体验）。“长尾为王”显示在矩阵的左上方，可以尝试提高该应用对其他市场的吸引力。

“杀手级应用”统治着移动行业。通过向大众市场中的用户提供高价值的应用，它不仅对使用应用进行收费，还能凭借其主导地位定期地收取广告费用。其他的应用不断渴望能成为杀手级应用，虽然多数永远都做不到。The Angry Birds（愤怒的小鸟）曾经是全球成功的案例，它提供付费版也提供有广告支撑的版本（当然，面向的移动平台不同）。

“百炼成钢”是很有前景的应用，有很大的市场潜力，但是在消费者眼里，这些应用并没有太多的感知价值。“百炼成钢”不得不非常努力地工作（由此得名），通过说服消费者接受其提供的价值找到一条出路。

开发者在决定对其应用收费或者在应用中包含广告（或者既收费又有广告）之前，需要自问下面几个问题：

- 我的应用有很多感知价值吗？
  虽然看起来显而易见，但是如果有很多竞争的应用也提供类似的功能，那么回答很可能就是否定的。如果这项应用是第一个提供新功能的应用，那么回答可能就是肯定的。
- 我的应用是不是打开了一个新的市场？
  如果是，就不能依靠一小撮用户产生大量的广告收入（本章后面会有更多介绍）。
- 应用中包含的广告是否会影响到消费者总体的用户体验？
  如果广告遮蔽了显示屏幕的很多部分，或者在莫须有的时刻突然冒出，就会很快惹恼用户，从而让他们丢弃该应用。
- 我能否提供足够的新功能来更新应用从而让用户为之付费呢？
  如果你觉得情形就是如此，那么一个精简的应用就能成为很好的开端。
- 应用的生命周期可能会有多长？
  对于花拳绣腿的应用，它的生命周期可能很短，因为消费者将很快地厌倦。在应用刚推出时收取较高的费用，然后再减少收费是很明智的做法。
- 我的应用是社交型的应用吗？
  对社交型的应用来说，任何费用都不可能在一段时间内持续收取。虽然先驱者比如 Loopt 能够保持长期收费的政策，但是现今的市场都希望这些应用能够免费使用。现在所有的 Loopt 应用程序都可以免费使用了。
- 我的应用能提供程序内购买吗？
  Apple 率先推出了小额付款平台，程序内购买不仅成为跨越不同开发平台的更受欢迎的收费方式，而且也变得更容易采用。它是重要的收入来源（见下一节），能真正替代以下载为基础的收费方式。

很明显，根本不存在通用的解决方案，即使某种收费机制看起来明显是好办法，由于市场时机、竞争应用以及其他因素的影响都可能造成不好的结果。在这个相对较新的领域，关键就是尽可能地采用多种不同的方法明智地进行实验，最后找到适合你特定应用的最好收费模式。

9.1.2　程序内收费

在 LBS 应用中对特定功能进行收费是一个给开发者很多期望但仍然很少有人触及的领域。

这就是所谓的程序内收费或小额付费（涉及的金额很小）。

Apple 于 2009 年 7 月发布了 OS 3.0，促进了用户的程序内购买行为，这也是此领域受到关注的部分原因。也就是说小额付费方式正慢慢地成为主流（已经成为 BlackBerry、Nokia Ovi 以及 Android 应用商店中必不可少的一部分）。另外一部分原因是因为固定网络和移动网络之间互不交叉。一般情况下，传统的网站是通过 PC 访问的，很多公司已经成功地引进了免费增值定价模型。在这里消费者可以免费访问一个网站的主要功能但是需要一次性付费或定期付款才能访问其附加的功能。

举一个例子来说明这个观点是很有用的。例如，领先的专业社交网站 LinkedIn 能提供免费会员服务，但是对额外的功能只有付费才能使用。订阅一个 Business Plus 需要花费 49.95 美元，订阅的会员可以从 LinkedIn 会员库中同时给 10 个其他会员发消息，可以查找 500 个个人简历。图 9-3 展示了由 LinkedIn 提供的付费订阅的服务范畴。该公司的定价策略非常成功，已经实现盈利并成功上市。

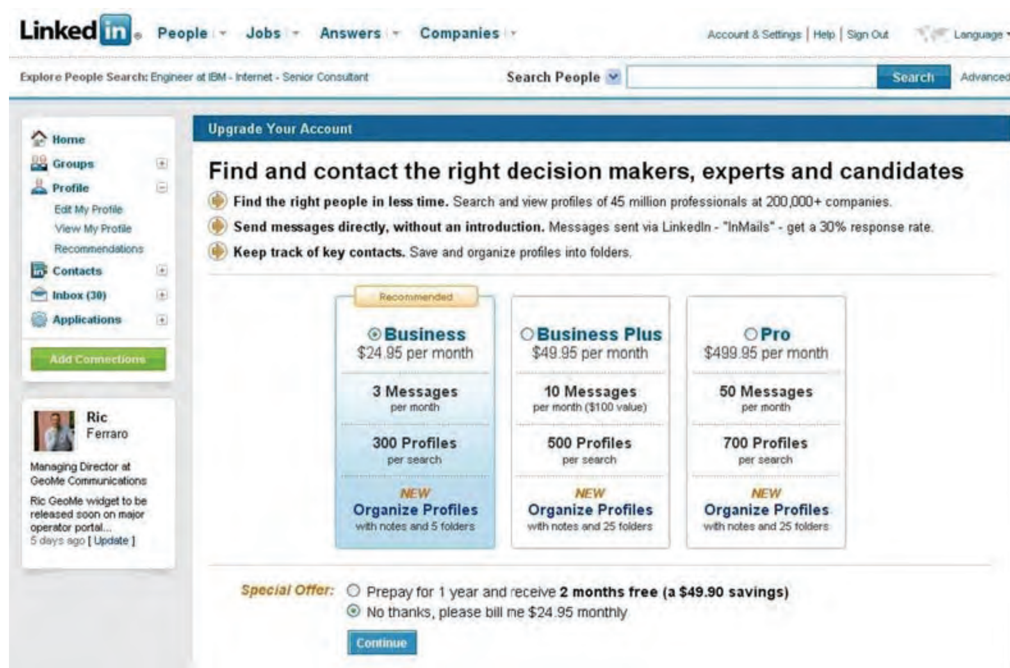


图9-3　LinkedIn在基本的免费会员制基础之上，通过提供三种不同价格等级的收费会员发展了成功的免费增值模型

另一个网络领导先锋、程序内购买的先驱就是 Habbo Hotel。Habbo Hotel 作为针对青少年的“虚拟世界的网络社区”是在 2000 年推出的，到 2007 年就已经取得了惊人的成功，拥有超过 7500 万的用户，建立了主要以虚拟礼物为基础的商业模式。虽然 Habbo Hotel 的主要思想是为青少年创建一个社交网络，但是其真正的意义在于创建了有关虚拟礼物和免费增值定价的整个货币化模

型。用户可以免费加入，但是要创建属于自己的虚拟世界，必须用 Habbo 钱币（以真正的货币兑换）购买虚拟物品（或用 Habbo 的话说教“furni”）。另外，用户能购买特殊技能，比如能够暂时“隐形”或者能制造一个“爆炸”效果。到 2007 年，Habbo 的全球收入达到了 6000 万多美元，销售虚拟礼物或特殊效果取得的收入就占了其收入的 90%。

今天，一个 Habbo Hotel 的翻版是 Farmville，一个实时仿真农场的游戏，2009 年 6 月作为 Facebook 的一款应用推出。这个游戏能让会员管理虚拟农场，能够种植、打理以及收获虚拟的农作物和树木，还能饲养牲畜。Farmville（如图 9-4 所示）有超过 5800 万的用户，它每天销售的虚拟拖拉机的数量达到了 50 万辆，此游戏一经推出就开始赢利了。



图9-4　Farmville的虚拟世界中，用户可以与社区会员联系，创建自己唯一的网上身份，并能利用网上商店中可用的农场工具设置自己的个性化虚拟农场

Habbo Hotel 倡导的免费增值模式，最早由英国的交友网站 Flirtomatic 践行。该公司创立于 2006 年，它们引人注目的在线情话及虚拟礼物都是免费增值模式的经典手法，为业界称道和效仿。自在线情话和虚拟礼物推出后的两年内，该公司通过向其 100 多万社区会员销售礼物，就有了超过 1500 万美元的收入。

Flirtomatic 的虚拟礼物有：简单的表示一下、含情脉脉的个人信息、“爱吻”，还有亲吻，这些“礼物”都可以用 Flirt 点数，即 Flirtomatic 货币购买。Flirt 点数可以通过信用卡或直接通过用户的电话账单购买。图 9-5 展示了用户发送消息以及利用 Flirt 点数购买虚拟礼品的手机屏幕截屏。虽然整个想法从表面上看比较无聊，但有一段时间此游戏在亚洲的虚拟商品中一直比较畅销，其

全球的包括在线和手机的销售收入大约有 15 亿美元之多[①]。

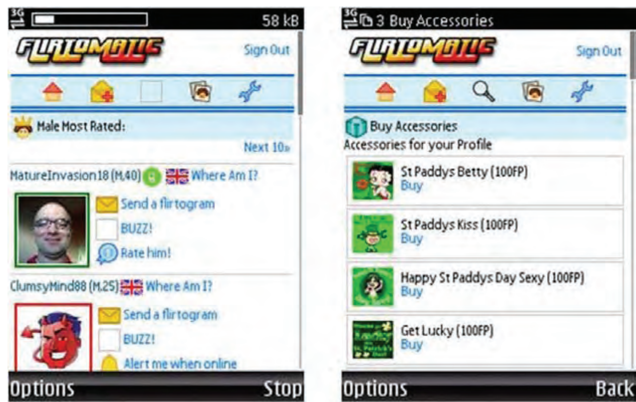


图 9-5　典型的 Flirtomatic 概况页面，用户可以购买不同的虚拟礼物

Apple 在 2009 年 8 月推出了自己的程序内支付平台，这个平台有很多特别的功能，包括开发者可以在应用中收费。由于 Apple 的 iTunes 商店中已有信用卡信息，因此用户更容易接受与其信用卡关联的小额付费方式。

这就意味着开发者现在可以对增值服务以及虚拟商品收费。例如，如果用户附近有特价优惠活动（通过图 9-6 所示的 iPhone Push Notification 提示功能）时，就可以告知社交网络上的用户，理论上这就是一次收费的机会。

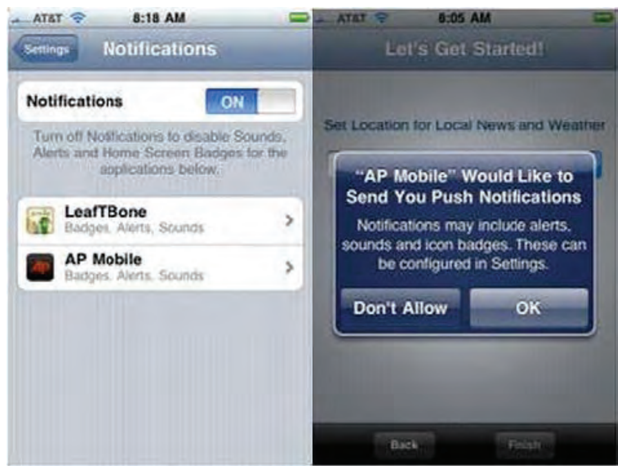


图 9-6　AP Mobile iPhone 应用程序中的推送通知。这是在 Apple 小额收费平台中对用户收费的新功能

① Susan Wu, Charles River Ventures, TechCrunch, 2007 年 6 月 20 日 (guest post); mng.bz/1P62。

Apple 的 iTunes App Store 可以识别三种类型的程序内购买：

- “消耗品”必须在每次用户需要时购买。例如一次性服务（比如访问更高级别的游戏）通常被当作消耗品。这就意味着如果用户是今天购买的，那么明天还要再次购买的话，程序将会对他再次收费。
- “非消耗品”的购买，一个特定的用户只需购买一次。一旦购买了非消耗品，它就能用于所有与用户 iTunes 账户相连的设备。Store Kit（连接 App Store 的 Apple 的软件框架，能保证应用中用户的安全支付，如图 9-7 所示）能在多种设备上提供非消耗品恢复的内置支持。
- “订阅服务”兼具消耗品和非消耗品的特点，是最复杂的一种类型。正如消耗品一样，订阅服务可能需要用户购买多次；开发者可以自己实现续费机制。然而，订阅服务必须能在可以连接到用户的所有设备上使用。程序内购买一般要通过外部服务器来交付。对开发者而言，当用户在所有与 iTunes 同步的设备中购买订阅服务时，必须确保任何一台设备中的订阅服务都是有效的。因此，不要将程序内购买锁定到某个设备标识。

有两点需要注意，第一，程序内购买不能用于产品更新。变化的二进制代码必须单独提交。但是如果你是一个游戏开发者，那么游戏数据、游戏地图、游戏等级以及其他的数据文件都可以用程序内购买实现交付。第二，每一个通过小额付款程序卖出的商品必须预先得到 Apple 的许可，就像 Apple 要审查每一个应用一样。

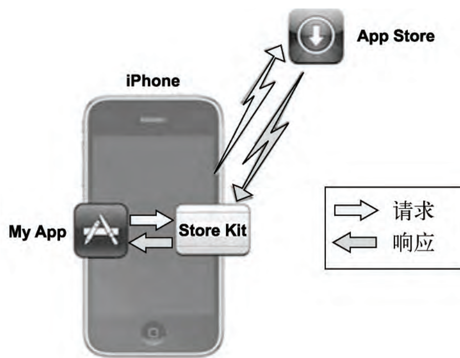


图 9-7　Apple 的 Store Kit 是连接 iPhone 应用和 App Store 的中介，通过展示可用的优质内容，对个人小额收费授权，让用户购买批准的项目（来源：mng.bz/qbJg）

要在 iPhone 应用中创建小额付款功能，首先需要在项目中添加 Storekit.Framework[①]。检测确认设备上没有父母限制后，就可以检索产品信息，填充用户界面。该功能通过几行代码就能实现：

① Mugunth Kumar, “iPhone tutorial—In-App Purchases,” October 18, 2009; available at mng.bz/30r5.

代码清单 9-1 检索产品信息并充填 UI

```
- (void) requestProductData
{
SKProductsRequest *request= [[SKProductsRequest alloc]          ❶ 创建请求对象并加以初始化
initWithProductIdentifiers: [NSSet setWithObject: kMyFeatureIdentifier]];
request.delegate = self;
[request start];                                      ❷ 启动请求
}
- (void)productsRequest:(SKProductsRequest *)request didReceiveResponse:
        (SKProductsResponse *)response
{                                                     ❸ 实现方法
NSArray *myProduct = response.products;
 [request autorelease];                               ❹ 存储收到的信息
}
```

这段代码展示了如何向产品来源请求产品信息，然后保存该数据。这段代码首先创建了一个请求对象❶，然后开始启动请求❷。如果请求成功返回❸，代码就会缓存返回的信息❹。

接下来需要设定一个观察者，即使支付程序被打断（例如，如果交易过程中有电话打进来），也可以从 iTunes App Store 中收到回调信息：

代码清单 9-2 创建一个交易观察者从 iTunes App Store 中接收回调

```
MyStoreObserver *observer = [[MyStoreObserver alloc] init];
[[SKPaymentQueue defaultQueue] addTransactionObserver:observer];
```

这段代码为该商店创建了一个观察者，并将它赋值给变量 `observer`。

为了执行回调并处理三种类型的事务（购买、失败以及恢复），只要打开应用程序，你就应该准备好接收这些通知。最好是通过 `applicationDidFinishLaunching` 或者一个等效的方法初始化这些程序。下面的代码就可以实现此功能。此功能适用于所有更新的事务，对每个事务都能处理。

代码清单 9-3 执行回调

```
- (void)paymentQueue:updatedTransactions(SKPaymentQueue *)queue
updatedTransactions:(NSArray *)transactions           ❶ 遍历数组中的每个事务
{
for (SKPaymentTransaction *transaction in transactions)
{
switch (transaction.transactionState)                 ❷ 要购买
{
case SKPaymentTransactionStatePurchased:
 [self provideContent: transaction.payment.productIdentifier];
break;
case SKPaymentTransactionStateFailed:                 ❸ 显示错误
if (transaction.error.code != SKErrorPaymentCancelled)
{
}                                                     ❹ 恢复应用，像购买过一样
break;
case SKPaymentTransactionStateRestored:
[self provideContent:
     transaction.originalTransaction.payment.productIdentifier];
```

```
default:
break;
}
 [[SKPaymentQueue defaultQueue] finishTransaction: transaction];  <-- ❺ 删除事务
}
}
```

这段代码展示了如何运行每个事务❶，处理所有不同的状态，购买状态❷或者失败状态❸，或者恢复状态❹。然后代码就会从等候队列中移除该事务。

购买和失败状态看起来很明确。如果在事务完成之前退出程序，你就会收到一条恢复事务的信息。如果购买行为是新的或者是恢复的，那么你做的事情应该一样。如果你想对用户的每一次下载都收费的话，那么就要将程序内购买设置成消耗品。但应该只针对实时广播或者播客这样用户真正消费的东西，而不能用于解锁附加等级。锁定的等级应该仍然保持锁定状态。

这里有三件重要的事情需要注意：

- 一旦事务完成就应该将事务从支付队列中移除。否则，程序将会再一次尝试，这可不是用户所希望发生的（否则，用户很有可能弃用你的应用）。
- 在交易完成之前你应该提供内容（或者解锁功能），当异步缴款通知完成时交易就成功了。如果你收到 SKPaymentTransactionStatePurchased 信息，这就意味着用户信用卡已经付费完毕了。
- 当交易失败时不能向用户展示错误信息，因为用户不喜欢。

完成程序的构建就可以继续下一步，当用户在 UI 里点击程序中的购买按钮时，调用下面的函数来初始化购买行为。

代码清单 9-4　初始化实际的购买行为

```
SKPayment *payment = [SKPayment
     paymentWithProductIdentifier:myGreatFeature1];
[[SKPaymentQueue defaultQueue] addPayment:payment];
```

这里创建了一个支付对象，并将它赋值给变量 payment。

现在你已经完成了准备工作，下面就是至关重要的开发阶段：对程序进行测试以确保程序内购买能够正确运行（要注意的是你必须直接在设备上做测试，因为此测试在模拟器上不能运行）。

开发者对小额付费系统充满憧憬，因为通过小额付费他们可以更好地控制以及灵活地处理程序内购买。有些批评者指出，不该让用户进行不到 1 美元的交易，因为会让他们觉得麻烦。开发者可以利用自己制造的货币（像 Flirtomatic 的 Flirt 点数）来避免这样的问题。这就意味着用户可以支付较大的金额，以便将来购买其他东西。

一次性付费也可以适用于应用中的优质内容。例如，像 weather.com 这样的供应商提供的当地天气预报可以作为基础应用中的一个附加图层提供给用户。

短信收费也是一种不错的一次性收费方法。这种方法可以让用户直接通过手机账单支付，开发商可以分享由此产生的利润。好处是操作简单（从用户的角度看）、使用可靠（这是久经考验的技术），缺点就是要依赖于移动运营商，并支付佣金（因为运营商会拿走大部分收入）。

短信收费是一种很好的一次性收费方法

有利方面包括：

使用方便——是所有计费机制中最容易的。通过发送一段简短代码的请求信息，移动用户就能在下次的手机账单中自动付费。

安全性——没有任何支付信息需要交换。

可靠性——已经实施了很多年（自从有彩铃收费以来），可以提供一个可靠的支付平台。

不利方面包括：

依赖移动运营商——依赖移动运营商（移动运营商会指定简短的代码并对客户开账单），需要签订协议。

经营者佣金——移动运营商会通过协议从内容供应商那里分得大部分利润，可以高达60%。

9.2 企业是收入的来源

尽管 LBS 的 B2B 市场还在初期，但却有着巨大发展潜力。此领域内的全球 LBS 订阅费用不仅早已达到 8.61 亿美元（2008）[①]，而且预计三年内会增加两倍多（见图 9-8）。另外，全球的手机广告投入早就接近 20 亿美元（2008），并有望在以后占有越来越大的份额。

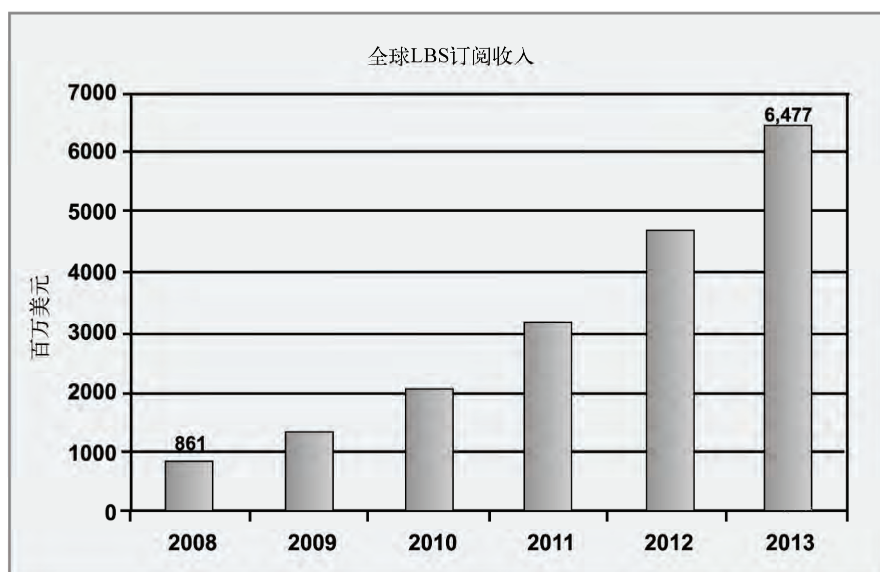


图 9-8 从 2008 年（实际的）到 2013 年（预计）的全球 LBS 订阅收入（百万美元）

① Informa,“Current State of the LBS market: Backdrop and Outlook,”Jamie Moss，2009 年 5 月 12 日。

谈到移动应用，从企业获取收入的方法有三个：

- 对应用中占用屏幕空间收费，或者对版面收费
- 对使用应用中收集的用户信息收费
- 对应用中的知识产权（IP）收（许可）费

9.2.1　版面费用

对于手机的版面，可以出租房子一样收费。问题是如何划分版面。对手机应用，有以下版面可供出租：

- 可以从手机上下载的网页
- 应用的初始画面
- 主应用屏幕或视图

可以插入手机横幅广告标语（比较常见的选择），也可以让广告客户买断整个应用（可能既要在应用中插入广告，还要展示其 LOGO 和简介）。可以参考图 9-9。



图9-9　版面的不同要素给应用开发商们提供了收费的机会（从左到右按顺序）：下载网页的应用、启动画面，包含了广告的主应用页面

决定了要出租哪块版面，下一步就是最有效地利用这块广告空间。为此，需要理解应用中可以包含的手机广告及规范。

手机广告

The Mobile Marketing Association（移动营销协会）定义了下列手机广告客户可用的主要可点击行为[①]:

- 点击致电——用户给内容供应商或广告客户打电话。
- 点击查找——用户可以利用基于位置的服务查询，例如最近的汽车经销商或者电影院。
- 点击订阅宣传册——用户通过提供邮寄地址收到宣传资料。

① “Mobile Advertising Overview,” 2009 年 1 月, 手机市场协会; http://www.mmaglobal.com/files/mobileadoverview.pdf。

- 点击参与竞赛——用户输入文本或抽奖号码赢取奖励。
- 点击接收邮件——用户通过提供邮件地址来接收邮件以及网站链接。
- 点击接收移动优惠券——用户在手机上收到电子优惠券，这些优惠券可以由参与的商家直接赎回。
- 点击购买——用户用信用卡支付或添加到每月的手机账单或者利用其他形式的手机支付方式进行购买。
- 点击下载内容——用户可以下载商标、壁纸或铃声到手机上。
- 点击进入品牌手机网站——用户点击广告图片访问一个长期有效的或者特定活动的移动网站。
- 点击转发内容——用户可以向朋友转发相关的内容，产生类似病毒营销的效果。
- 点击视频——用户点击广告图片观看商业广告。
- 点击投票——用户用手机投票，给市场和品牌研究公司提供宝贵的建议。

在移动应用中可以包含一个或多个这样的可点击行为。你必须考虑的一个关键因素就是点击行为是否会把用户带到移动应用外部（例如在 iPhone 应用中点击一个网站链接）。如果这样，必须要权衡潜在的广告收入以及这种广告对用户体验产生的不利影响。很明显，较少的广告打扰会使整体效果更好。

最后，开发者的目标就是让应用中的每个广告都尽可能地获得较高的点击量或者浏览次数（取决于广告是基于 CPC 还是 CPM），因为这会决定广告收入。

CTR（点通率）和 CPM 是衡量一个广告活动效果的常用方法。广告发布方（应用的所有者）收到的广告收入是根据广告爆光次数，或广告的点击次数计算的。

CPC 及其他点击付费广告通常是利用竞标机制计算的，不同广告活动的费率存在一定的差异，但通常都低于 1 美元。CTR 手机广告收入比 Web 广告收入要高一些，平均能达到 5%左右。在针对性更高的广告活动中（例如，与旅游有关的应用，广告客户是一家航空公司），收入会更高。

展示手机广告的更多实用示例可以在全球移动广告领导组织美国移动营销协会实操文档中找到，下载地址为 www.mmaglobal.com/bestpractices.pdf。

发布手机广告最常用的办法就是利用第三方的移动广告网络，因为它比自行开发一个解决方案要快得多。最近几年，全世界的手机广告经营商如雨后春笋般涌现，最知名有 AdMob、Smaato 和 Admoda。

在手机应用中展示广告很简单，访问移动广告网络的服务器，即可获得广告文本或图表。代码清单 9-5 展示了如何在你的手机网站上插入 AdMob 广告，而代码清单 9-6 到 9-9 展示了如何在 Android 应用中插入 AdMob 广告。

代码清单 9-5 利用 Python、Django 在手机网站上插入 AdMob 手机广告

```
from urllib2 import urlopen
from urllib import urlencode
```

导入必要的类

```
try:
    # Python >= 2.5
    from hashlib import md5

except ImportError:
    # Python < 2.5
    import md5                              根据版本导入 md5 类

def admob_ad(request, admob_params=None):            ❶ 实现主函数

    """
    Given a Django "request" object and dict of
    admob parameters returns a Admob ad.

    If no ad can be retrieved displays a one pixel

    Admob tracker image.

    Usage:

    def my_view(request):

        admob_dict = {}
        admob_dict["admob_site_id"] = "required_admob_site_id"
        admob_dict["admob_postal_coode"] = "optional_postal_code"

        admob_dict["admob_area_code"] = "optional_area_code"
        admob_dict["admob_coordinates"] = "optional_coordinates"
        admob_dict["admob_gender"] = "optional_gender"
        admob_dict["admob_keywords"] = "optional_keywords"
        admob_dict["admob_search"] = "optional_search"

        ad = admob_ad(request, admob_dict)

        return HttpResponse(ad)

    """

    admob_mode = "test"                      准备好部署时，改为“实时”

    admob_endpoint = "http://r.admob.com/ad_source.php"

    admob_version = "20080714-PYTHON"
    admob_timeout = 1.0
    admob_ignore = ("HTTP_PRAGMA", "HTTP_CACHE_CONTROL",
    ➥ "HTTP_CONNECTION", "HTTP_USER_AGENT", "HTTP_COOKIE",)

    admob_post = {}            ❷ 开始构建 URL

    admob_post["s"] = admob_params["admob_site_id"]      声明必要的参数


    admob_post["u"] = request.META.get("HTTP_USER_AGENT", None)      声明元参数

    admob_post["i"] = request.META.get("REMOTE_ADDR", None)
```

```
    admob_post["p"] = request.build_absolute_uri()

    admob_post["t"] = md5(request.session.session_key).hexdigest()

                                                      声明硬编码的参数
    admob_post["e"] = "UTF-8"                       <-

    admob_post["v"] = admob_version

                                                           声明可选的参数
    admob_post["ma"] = admob_params.get("admob_markup", None)  <-

    admob_post["d[pc]"] = admob_params.get("admob_postal_code", None)

    admob_post["d[ac]"] = admob_params.get("admob_area_code", None)

    admob_post["d[coord]"] = admob_params.get("admob_coordinates", None)

    admob_post["d[dob]"] = admob_params.get("admob_dob", None)

    admob_post["d[gender]"] = admob_params.get("admob_gender", None)

    admob_post["k"] = admob_params.get("admob_keywords", None)

    admob_post["search"] = admob_params.get("admob_search", None)
for k, v in request.META.items():

    if k not in admob_ignore:
        admob_post["h[%s]" % k] = v

                                          去掉空值
for k, v in admob_post.items():         <-

    if v is None or v == "":

        admob_post.pop(k)

 if admob_mode == "test":

     admob_post["m"] = "test"

                                              ❸ 设置 GUI 元素中的文本
admob_success = True
try:                                        <-
    admob_data = urlencode(admob_post)

    admob_file = urlopen(admob_endpoint, admob_data)

    admob_contents = admob_file.read()
    if admob_contents is None or admob_contents == "":
```

```
        admob_success = False
except Exception, e:

    admob_success = False

if not admob_success:

    admob_contents = "<img src=\" http://t.admob.com/li.php/
    ➥c.gif/%(admob_site_id)s/1/%(admob_timeout)/%(absolute_uri)s
    ➥\" alt=\"\" width=\"1\" height=\"1\" />"  \s

        % {"admob_site_id" : admob_params["admob_site_id"],

             "admob_timeout" : admob_timeout,
             "absolute_uri" :
 md5(request.build_absolute_uri()).hexdigest()}


# print 'Connecting to: %s' % admob_endpoint        <-- 开始调试
# print 'Sending Parameters:'
# print admob_post

# print 'Got reponse:'
# print admob_contents

return admob_contents
```

我们给主函数的实现❶传入 Django 请求对象和 AdMob 参数词典，它返回一个 AdMob 广告。此代码开始创建了请求 URL❷，在添加完所有的参数后，代码就向服务器❸发出请求，将收到可以显示在屏幕上的 AdMob 广告。如果检索不到广告，代码就会显示占位符。这一点很重要，因为如果你的广告区域是空白的，那么页面可能就会出问题。

知道了如何在手机网站上插入广告，再看一看代码清单 9-6 至代码清单 9-9，它们展示了如何在你的 Android 应用中插入 AdMob 广告。

代码清单 9-6　创建一个新的活动

```
package org.ifies.android;                          导入必要的类
import com.admob.android.ads.AdView;
import android.app.Activity;
import android.os.Bundle;                              ❶ 声明主活动类
import android.widget.TextView;
public class AdmobExample extends Activity{

  @Override
  public void onCreate(Bundle savedInstanceState) {     ❷ 取得 GUI 元素的指针
    super.onCreate(savedInstanceState);
    this.setContentView(R.layout.admob_example);
    example_message = (TextView) findViewById(R.id.example_message);
```

```
    example_message.setText("This is an example of AdMob
    ➥for Android");
    example_adview = (AdView) findViewById(R.id.ad);
    example_adview.setVisibility(AdView.VISIBLE);

  }
  private TextView example_message;
  private AdView example_adview;
}
```

❸ 设置 GUI 元素的文本

❹ 显示元素

这段代码展示了如何将 AdMob 广告元素插入到 Android 应用中，以及如何显示广告。首先声明主要活动类❶，然后取得 AdMob GUI 元素的指针❷。接着设置此 GUI 元素的文本❸并使其可见❹。进行错误检查并知道哪里没有返回 AdMob 广告很重要。用户安装应用后，该应用就会发出请求，在设备上获取某些特权，比如访问设备的位置或者连接设备上的照相机。在这种情况下，我们必须将 AdMob 发布者的 ID 作为一个配置项添加到应用中。现在必须要创建一个视图，如下面代码清单所示。

代码清单 9-7　创建一个能显示广告的视图

```
<LinearLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res/org.ifies.android"
android:orientation="vertical"
android:layout_width="fill_parent"

android:layout_height="fill_parent"
android:layout_gravity="bottom">
```

创建好这个简单的视图后，就需要创建 attars.xml 了。在文件中，我们将声明新的 styleable。

代码清单 9-8　在 res 目录内，创建一个叫做 attars.xml 的文件

```
<resources>
    <declare-styleable name="com.admob.android.ads.AdView">
    <attr name="backgroundColor" format="color" />
        <attr name="textColor" format="color" />

        <attr name="keywords" format="string" />
        <attr name="refreshInterval" format="integer" />
        <attr name="isGoneWithoutAd" format="boolean" />
```

最后，我们会为你展示如何修改 XML file 证明。

代码清单 9-9　在代码中加入许可及发布者的 ID

```
<meta-data android:value="YOUR PUBLISHER KEY"
     android:name="ADMOB_PUBLISHER_ID" />
```

怎么样,利用广告网络特定的程序库在应用中插入手机广告很容易吧,此例中使用的是 AdMob Android 程序库。

值得指出的是，有少量主要的移动广告网络声称有能力发布基于位置的广告（如通过 LBS 应用，根据位置动态改变广告），尽管调查显示主要的移动订阅者愿意在 LBS 应用中收到当地的

广告信息，尤其是能够链接到商铺打折促销信息的广告，但还没有人大规模的推出此项服务。

新的位置基础设施供应商正开始建立高度本地化的广告服务，进军由主要移动广告网络留下的这个新市场。例如，像 Placecast 公司在美国提供了一种基于位置的广告网络，而其他的如 Maporama（法国）在欧洲与移动运营商一起捆绑提供这种服务。

Nokia 通过其 Ovi Maps 服务，现在也能够基于 Nokia 设备获得的实时位置为广告客户发布广告。例如，下面的图 9-10，就是用嵌入式广告在附近的地铁出口展示 Helsinki（赫尔辛基，芬兰首都）当地的天气预报。



图 9-10 Nokia 的 Ovi Maps 截屏，展示了基于位置的广告怎样嵌入当地的天气预报服务。用户点击广告标识，就能看到一个商铺的详细信息及其在地图中的确切位置

另一个例子是日本的 Otetsudai Networks（图 9-11 所示）在 2008 年推出的一项服务，能利用短信和手机上的 GPS 将雇主的短期临时任务发布给找工作的人。

比如一个饭店在周六晚上缺人手，需要找人用一个小时刷洗盘子。饭店经理通过手机登录到 Otetsudai Networks，填写了一份简单的标准表格，然后点击发送。Otetsudai Networks 马上就能派发出上百条短信给距离此饭店不远的人，通知他们查看此信息。几分钟内，经理收件箱里就会收到潜在雇员的响应信息——资格条件，等级评定，甚至有时会有一张照片以及表达个人兴趣的信息。如果候选人员的费用不合适，经理就可以弃用。如果找到合适的人选，可以点击雇用。雇主要通过信用卡向 Otetsudai 支付相当于该工作薪水一半的寻找费用；而雇员仅需要支付登录该工作网站[①]的传输费用。

① Lisa Katayama,“Tokyo upstart offers freeters mobile flexibility,”Japan Times, June 4, 2008;链接为 mng.bz/e67Z。

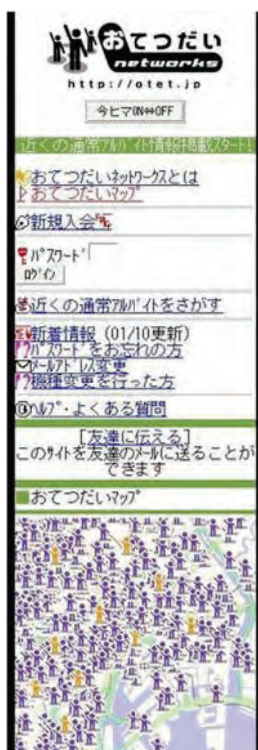


图 9-11　Otetsudai Network 革命性移动应用的屏幕截图，通过手机上的 GPS 让雇主和找工作的人互相发现对方

9.2.2　位置数据收费

随着一般的 LBS 应用数量不断增加，不同移动平台各种竞争产品的推出，很多公司不得不决定挖掘 LBS 更深层次的价值。尤其在将来，不仅应用本身具有价值，就连应用中由用户捕获的大量位置数据也有价值。

Google Maps 就是一个很好的例子。Google Maps for Mobile 是其地图软件的手机版，最初用于粗略地（有时定位误差能达到几公里）确定手机用户的位置。这是由于这种定位方法要依靠于基站三角测量（见第 2 章），但又没有基站准确位置的最新数据库的缘故。随着用户越来越多，Google 收集了每一次用户与其服务联系时的小区基站 IDs 的位置数据。经过几年的数据积累，Google 已经建立了世界上最全的小区识别码数据库。

尽管 Google 让终端用户免费使用这些数据，但是会间接向企业收取费用。这是由于 Google Maps for Mobile 要依赖于从用户那收集的数据进行定位，而企业就要为此付费。这些数据也能让 Google 完全绕过移动运营商，能够独立的进行位置探测，并且是免费的！

另外一些公司将自己定位为位置数据处理商，获取和处理不同来源的数据，然后向第三方销

售其数据。美国 Sense Networks 就是一个例子。

Sense 于 2008 年在美国旧金山推出了一个概念验证消费者应用，Citysense。在手机上下载了此软件的订阅者必须同意网络追踪他们的位置，并将其加入一个“部落”中，或者志趣相投的一个组里。如图 9-12 所示，由深红色表示人员密度高。如果某“年轻又前卫的”用户（在应用显示为一个红色的圆点）在凌晨 1 点想找人陪一陪，她可以在手机上打开城市地图，看看哪个俱乐部显示为红色圆点的朋友。



图9-12　Sense NetWorks的黑莓应用中展示的“热区”表示有用户在那里聚集。这样就可以针对用户群投放广告

然后 Sense 专注于将移动数据细化用于商业，包括广告客户。它能展示出特定的“部落”是怎样聚集在一起的，在旧金山哪里有啤酒促销，并且说服广告客户将促销活动推广到该市的其他酒吧，这样就能吸引相同颜色的圆点集中起来。因为大品牌公司需要针对手机用户开展营销活动，这样 Sense 就能通过向移动运营商和这些公司收取费用将其服务转化为财富。这样 Sense 就迈进了这块由主要移动广告网络开辟的领域，而当前的移动广告网络并不愿提供有针对性的基于位置的广告。

值得注意的是一些基于位置的移动社交网络可以让用户评选他们喜欢的地点（位于纽约的 Centrl 就是很好的一个例子）。能挖掘这些地点的价值，并能向各大品牌和广告客户提供建议的公司都有巨大发展潜力。但相关数据以及销售对象则不局限于广告商。

实际上，有更多的企业迫切需要位置数据。特别是零售商们更希望获得这类信息。例如，美国的一家叫 Johnny's Lunch 的专营热狗的快餐店，在转型为特许经营时，就借助了大量位置信息。位置信息在任何公司的计划执行及规模扩张方面都扮演着至关重要的角色，能帮公司找到目标客户和理想的市场。[①]

① Matt Marshall, “Sense Networks gets $6m in hotly contested deal for ‘tribe’ advertising,” SocialBeat, June 26,2009; mng.bz/e67Z.

9.3 把知识产权转化为财富

本书第 12 章将会介绍知识产权（IP）的问题，讨论开发者如何保护自己的投资。但是不要忘了，应用中的 IP 也能为开发者带来利润。有两种方法可以将知识产权转化为财富，那就是许可和贴标签。

9.3.1 许可

第三方的应用许可通常表现为技术许可协议。若要许可他人使用，首先需要成功取得专利，而美国的开发人员常常比其他国家的开发人员更容易为自己的软件申请专利。更多的细节信息可以从 World Intellectual Property Organization（世界知识产权组织，WIPO）上查到。

许可的优势在于你拥有并能控制知识产权，而且能通过获得持续的版税收入来最大化商业回报。其不利的方面就是你需要维持、监督以及执行该项权利。

通常情况下，许可证分专有许可和非专有许可两种。如果你授予第三方专有许可，这就意味着只有这个第三方将有权在许可有效期内行使该知识产权对应的条约细则规定的权利。正因为如此，相较于非专有的许可，专有许可版权费用更高。

另一方面，如果你授予他人的是非专有许可，则意味着你可以继续授予其他第三方非专有许可。

在进行许可谈判之前，你应该考虑下面几点：

- 你是否考虑专有许可。
- 价格以及付款方式。
- 许可授权的地理范围有多大。
- 许可有效期有多长。
- 被许可人能否再许可他人。
- 如何处理所有权和改进的使用权（与专利许可相关）。
- 对许可证颁发者而言，持有者应该提供多少技术支持，需要花多少钱。因为这是颁发许可的主要目的，因此需要在谈判过程中花费较多时间。
- 持有者的市场义务以及颁发者的市场支持承诺。

如果你决定采用许可的方式，应该找一位专业的商标及知识产权代理。

9.3.2 贴标签

白标（white label）产品或服务是指一家公司（生产者）生产的产品或服务，由其他公司（商家）重新贴牌，使其看上去好像是后者制造的一样。尽管共享移动应用或服务器（用于处理应用发出的请求）也是一种方案，但推荐的方法是使用 API 模式。使有 API，只要向解决方案的购买者披露某些信息就可以，因此可以保护应用的知识产权。

例如，GyPSii 向手机制造商提供了 GyPSii Open Experience API。目的是让这些制造商把基

于位置的社交网络功能整合到手机客户端和手机应用中。实际上，GyPSii 的解决方案是供市场免费使用的。



图 9-13　iPhone City Guide 应用中 Lonely Planet 的范围截屏，目前能为世界上不同的城市提供 20 个不同的版本

高端旅游内容供应商（类似图 9-13 所示的 Lonely Planet City Guides）是白标 LBS 解决方案的新市场。他们能对应用（例如，iTunes 商店中每一个 Lonely Planet iPhone City Guide 定价约为 20 美元）收取较高的费用，并给合作开发人员提供良好的收入机会。

9.4　小结

LBS 给开发者提供了巨大的机遇，将他们带入位置感知应用和服务这个刚刚起步的市场。B2C 市场是很多开发者掘取第一桶金的始发站，在尝试收费或者冒险进军 B2B 市场之前，他们能在这里验证新的观点，测试新的需求。从一次性收费到自由增值，再到广告，从移动应用中挣钱的方法比以往任何时候都多。获得成功的关键就是选对方法（分析应用的市场潜能，明白应用的感知价值），从错误中总结经验，根据市场反馈做出调整。

在这个日新月异的环境中，保持灵活是开发者生存的首要法则。对手机广告来说，更是如此，这里为开发者发挥屏幕空间价值提供了空前的可能性。尽可能多尝试一些营销手段和可点击的行为，随着时间推移，相信你的付出一定会有回报。

第 10 章

隐私之争

10

本章内容

- 什么是隐私
- 关于隐私的两种看法
- 谁在控制位置感知应用中的隐私
- 隐私相关的法规

在第 9 章讨论如何通过位置感知应用和服务赢利后，我们开始新的一章。我们知道，位置感知应用和服务如何通过都是由普通大众掌控的，但人们对应用和服务中的隐私问题也一直在争论不休。

如果你向移动专家和 Web 开发专业人士请教，什么是影响 LBS 快速发展的最大障碍，我敢打赌他们的回答将会是“隐私”

隐私和位置被越来越多地一起提到（试着将这两个词放在一起搜索，会得到超过 19 亿条相关的信息），而这背后的原因就是相较与其他类型的隐私（宗教隐私、文化隐私、行为隐私及其他隐私），人们更看重位置隐私。

正因如此，任何 LBS 开发者或企业主都需要特别关注消费者对隐私的态度：

- 什么是普通大众所担心的？
- 如何减轻他们的担忧？
- 隐私有价吗？

在回答上述问题的同时，LBS 拓荒者们需要明白，面对新技术侵入，公众存在一定程度的隐私恐惧是正常的。这也就使得公众（和你的消费者们）的隐私教育变得很有必要，开发者必须恰当地告知人们的权利以及如何保护他们的隐私。

但是只有在很好地理解了隐私的含义之后，你才能这么做，尤其对于位置隐私而言。而本章就要为你摸清变幻莫测的隐私争论的底细提供必要信息。如果你在开发一个针对消费者的应用，有了这些信息，至少能使你的应用满足隐私法规，并且能避免一些令人讨厌的罚款。通过改善服务来减轻人们的隐私恐惧，或许就能在消费者市场中开创一片天空。

10.1 到底什么是隐私

在早期的 LBS 中，隐私就像个刹车。这也许是隐私争论中唯一没有任何人质疑一个共识。这个共识被个人隐私会受到“老大哥”监视的臆想所加剧，某种意义上，人们偷窥的普遍本能也强化了这一点（看看《老大哥》、《诱惑之岛》等电视剧有多受欢迎就知道了）。

但是隐私对我们来说到底意味着什么，为什么这么多人担心会失去它？

隐私的定义

隐私就是一个人能掌控自己个人信息的能力[①]。侵犯个人隐私可被视为对个人控制个人信息自由的限制。隐私也可以被看作“一个人独处的权利”或者人们访问其他人的个人事件和信息会受到限制的情况[②]。

隐私担忧与保证个人资料的机密性有关，与个人认为有可能泄露机密的潜在风险有关。在极端情况下，位置信息处理不当将会使个人处于危险之中，或严重地危害他们的社会生活或财产安全。

由于隐私的概念很广，一些观察者为与位置相关的数字服务定义了隐私的概念。而 2009 年出现的位置性隐私（或位置隐私）被表述成

个人在公共场所活动时，在正常环境下，他们的位置不会被系统地秘密记录以备后用[③]。

通过管理位置隐私，你就能够保护个人不会接受位置数据库的询问而回答下列问题：

- ❑ 周二你去过反战集会吗？
- ❑ 你流过产没有？
- ❑ 你看过艾滋病医生吗？
- ❑ 你在午餐时间是否去过汽车旅馆登记入住？
- ❑ 你是那个向安全机构匿名举报机器老化的人吗？
- ❑ 你会去哪个教堂？哪个清真寺？哪个同性恋俱乐部？
- ❑ 谁是你约会的前女友？[④]

一些研究已经确定了隐私的更多细节：隐私信息的关注点（The Concern for Information Privacy，CFIP）理论是由 Smith 等人[⑤]提出的，从四个方面确定隐私的关注点：

(1) 收集数据量，反映对个人数据收集及存储的广泛程度。

(2) 未经授权的二次使用，反映了对收集的用于某种用途的个人信息，却在未经许可的情况

---

① W. A. Parent,“Privacy, Morality, and the Law,”in *Philosophy and Public Affairs* vol. 12, no. 4 (1983):269–88。

② Philip Brey, editorial introduction,“Surveillance and Privacy,”in *Ethics and Information Technology* 7, no. 4: 183–84。

③ Andrew J. Blumberg and Peter Eckersley,“On Locational Privacy, and How to Avoid Losing It Forever,”August 2009,available at http://www.eff.org/wp/locational-privacy。

④ 同上。

⑤ H. J. Smith, J. S. Milberg, and J. S. Burke,“Information Privacy: Measuring Individuals’ Concerns about Organizational Practices,”*MIS Quarterly*, vol. 20, no. 2 (1996): 167–96。

下用于其他用途的担心。

(3) 错误，反映了人们对有意无意地不恰当使用个人数据的担心。

(4) 不恰当的访问，反映了对未经授权的人们可以顺利浏览或使用个人数据的担心。

图 10-1 阐述了这四个关键的隐私关注点。

开发者和 LBS 供应商在进军 LBS 服务领域时，为了避免隐私问题就要考虑这四个方面。现在你已经知道了隐私意味着什么及其关注点，接下来我们讨论围绕隐私都有哪些争论，以及一些更具体的问题。

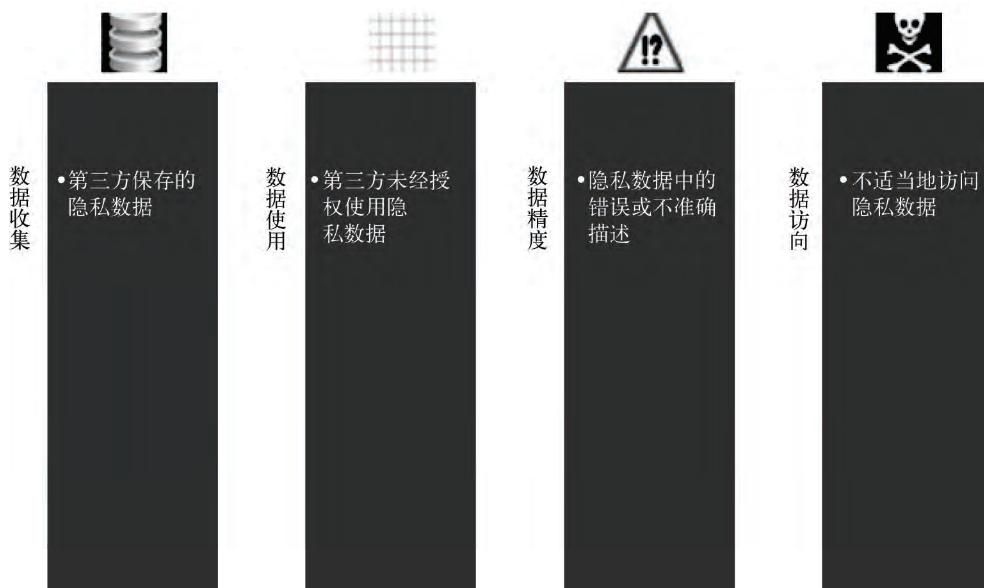


图10-1　隐私的四个担忧方面（为Smith等人的CFIP工具采用），包括了数据收集、数据使用、数据精度、数据访问，总结了当前的普通大众对隐私的担忧

10.2　隐私之争

有关隐私的争论并不是最近才出现的，但数字革命显然起到了火上浇油的作用，在全球引起了涉及社会、经济以及政治的广泛讨论。在本节中，我们将会了解什么导致了隐私的争论以及现今存在什么样的隐私威胁。我们将会介绍通过检测用户确切位置自动地向手机用户推送信息的技术。我们也将了解如何通过双向选择让用户获得隐私控制权，而这也将是化解争议的有效方法。

有关隐私的问题实在太多了（其中有很多至今没有答案），无数利益冲突导致的争论让相关问题还在继续增加：

- 谁来为隐私负责？
- 要由个人或政府来管理隐私吗？

- ❑ 什么时候能够容忍放弃隐私？
- ❑ 什么时候以及什么情况下应该通知个人可能会涉及其隐私？由谁通知？
- ❑ 能否给隐私定价？
- ❑ 为了尊重人们的隐私权，在什么地方应该划出道德和商业底线？
- ❑ 虚拟世界中的隐私保护应该与现实世界中的享受同等待遇吗？
- ❑ 根据个人的实时位置提供的某些服务是市场所能接受的吗？

更多围绕隐私的话题不仅是个人信息被收集的问题，而是它发生的太普遍、太安静、太廉价。很明显，除非你隐居在荒岛上，否则完全的隐私是不可能存在的，这也许能减轻公众因为被收集和处理个人信息而受到的惊扰。实际上，在这个世界的现在和将来，无处不在的设备和应用正安静地收集着人们的隐私信息，很多团体能查询、购买，或者直接调阅它们——甚至能雇用黑客盗取每个人的历史位置信息[①]。

现在我们将了解一些用户要面对的隐私威胁，以及违反哪些安全规定会影响到用户。

10.2.1　隐私威胁

CFIP 理论强调的四个隐私关注点可以转变为各种各样的隐私威胁，这些威胁可以划分为下面几个大的类别：

- ❑ 发送垃圾邮件——个人信箱中大量不请自来的信息
- ❑ 钓鱼网站（网络欺骗）——尝试获得用户名等敏感信息的欺诈行为
- ❑ 身份盗用——通常为了获取资源，通过假冒某人的身份，一些人装作是另外一些人的欺诈形式
- ❑ 不为人知的政府使用——例如政府机构为验证应税身份要使用
- ❑ 个人资料的恶意使用——竞争者、潜入者、欺小凌弱者使用

图 10-2 对比了这些不同的违反安全规则的例子（或者隐私事件），以及它们将会给个人带来的影响[②]。这有助于理解即使是看似无害的网上跟踪活动（比如网上银行），图表上的点 A，都会使个人面临重大的风险（比如未经授权的银行转账）。

这也暗示着即使是使用集合的个人档案（图表中的点 B）也会给个人隐私带来巨大的风险，因此开发者在向相关各方当事人交付 LBS 服务时，需要努力应对、严格审查和左右权衡[③]。

Google Street View（街景）是招致这种隐私争议的典型案例，如图 10-3 所示（该服务能提供全球各地 360 度街道视图）。该服务从 2008 年开始在不同国家相继推出以来（从美国到欧洲到日本），不断遭到公众对隐私侵权行为的强烈反对，偶尔还被告上法庭。

---

① Blumberg et al.,“On Locational Privacy”.

② David Riphagen, Probability Impact Matrix of Privacy Incidents, October 23, 2008.“The Online Panopticon.Privacy Harms for Users of Social Network Sites,”3TU (TU Delft, TU Eindhoven, and University of Twente),Centre for Ethics and Technology.

③ David Riphagen,“Privacy infringement—Directions for protecting users’ privacy online,”June 25, 2007.

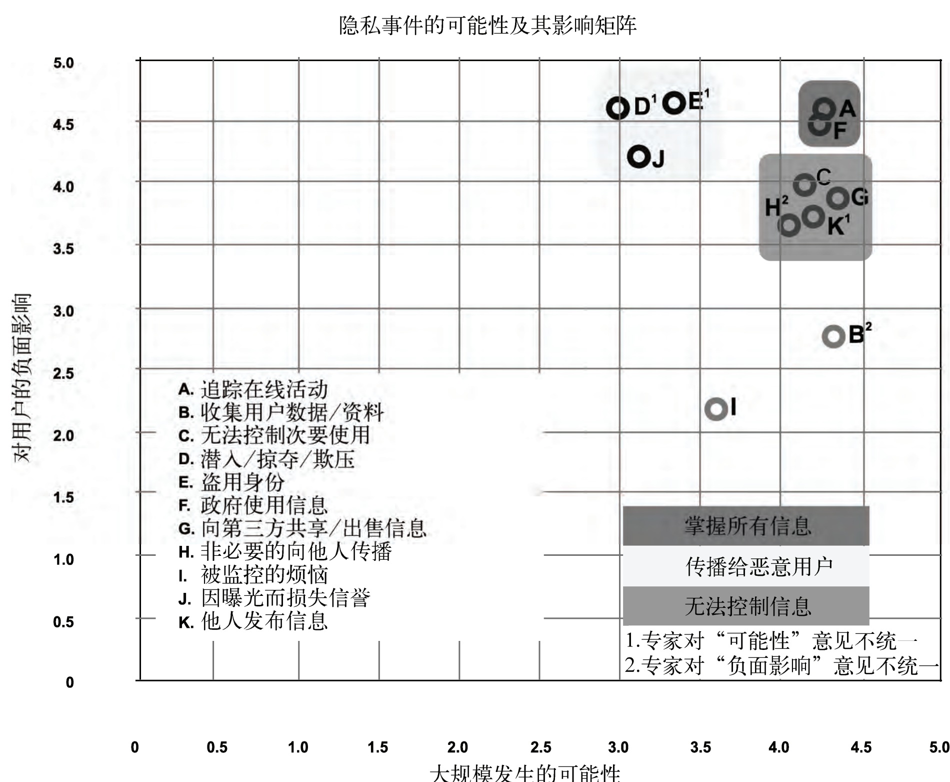


图10-2　隐私事件的可能性及其影响矩阵，展示了不同隐私侵犯发生的可能性及其负面结果（从次要的到严重的，比如身份盗用）。（来源：经David Riphagen许可转载）

对个人而言，他们尤其担心自己的私人住宅照片被所有人看到。在某些情况下，包含在 Street View 中突然闪过的面孔都能被识别出来——你不仅能知道一些人的私人住宅，甚至能知道居住在那里人的容貌。

谷歌公司的观点简洁地展示在美国的一份诉讼文件中：

完全的隐私根本不存在。

虽然在提供该服务时，谷歌公司并不违反任何法律，但是有关隐私的激烈争论反映了人们对此服务的强烈反应。一个叫 Osamu Higuchi 的评论者（他自己是一个专业的 IT 人士）对日本的 Street View 功能进行了强烈的批评，这表明在决定潜在的侵入服务是否能够被容忍中，文化起着很重要的作用：

在我们的生活方式中，你不能单方面，以机器可读的形式，将人们的生活空间向整个世界开放。

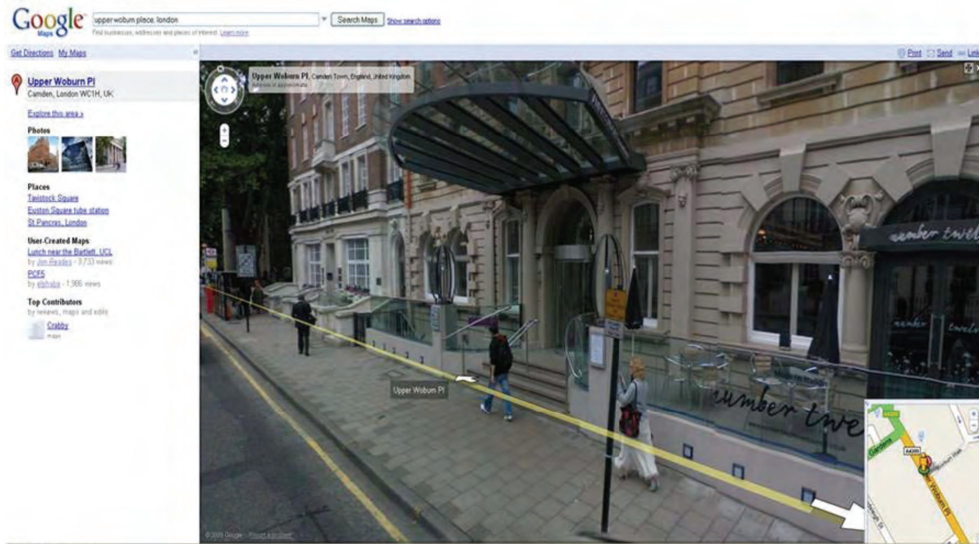


图10-3　Google Street View（Google Maps的一部分）能显示世界上（从日本、美国以及欧洲）一些国家的街景快照，但是该服务激起了侵犯个人隐私的争论，因为在特定的地点，该服务能描绘出一个真实的人

其他评论者则抱怨说尽管 Street View 不管任何人走过街道，都呈现出相同的图像，但谷歌是从一个 2.5 米高的人的角度向整个世界呈现视图，而这么高的人在现实中是不存在的。

尽管对该服务的反应程度会由于文化的不同而有所差异，总的来说，谷歌不得不多次被迫地按政府要求隐藏人物面貌、建筑物或场所。也许是因为谷歌点燃了隐私争论的战火，全球社交网络巨头 Facebook 和 MySpace 最初限制在他们的服务中推出附加的位置功能（但 2010 年 Facebook 在美国然后在欧洲相继推出了 Facebook Places 功能）。对于提供基于位置社交网络的初创企业而言，这是一个好消息，Foursquare 和得克萨斯州的 Gowalla（在第 5 章中介绍过）就因此迅速地发展起来。

现在你已经知道了个人可能要面对的由未经授权的隐私资料访问带来的特定隐私威胁，接下来将会特别介绍为什么推送位置的方式会让用户额外担忧。每个问题都有相应的解决方法，因此我们将了解选择加入（opt-in）方法，这是一种有效地缓解位置隐私恐惧的方法。

10.2.2　位置推与拉

正如第 1 章讨论的，隐私立法能够区别两种信息网络，比如手机服务的订阅者，有人会通过选择加入（因此同意展示其位置）要求使用位置感知服务，而有人则可以选择离开。这种行为差异是很重要的，因为 LBS 相关的隐私问题大都涉及在用户不知情的情况下被追踪位置。

尽管从表面上看选择加入是解决 LBS 隐私问题的一个便捷方法，但这种方法也只能解决问题的一部分。对那些采取不合适的措施保护自己隐私的人群来说，积累的问题很多：对一般消费

者而言，要理解有哪些办法可以避免自己的位置被记录下来常常很难，而当消费者与新的 LBS 服务互动时，要研究清楚这些问题也不容易。在当今的数字时代，人们并不能足够快速地适应先进的技术，以便直观地选择正确的方法。

订阅者选择加入与否，决定了 LBS 所能递送的服务类型：

- 主动使用，或者拉动——消费者的要求使用位置信息。典型的例子就是与位置相关的信息（天气，本地搜索）和导航。
- 被动使用，或者推送——位置请求不是由消费者发起的。典型的例子就是寻找好友和定位追踪。营销公司也可能利用推送服务根据个人的实时位置在某些地点提供某些服务，如今对 LBS 恶意行为的担心对这种做法有所限制。具有讽刺意味的是，至少在理论上，把位置作为一个附加的目标参数，广告商们将可以或多或少地发送一些相关商业信息，使终端用户和广告商都受益。

选择退出的订阅者就不能使用推送服务了，因为此服务只能在允许追踪其位置的手机上运行。

选择进入与选择退出

选择进入是当今 LBS 服务中管理隐私的基本功能。隐私立法已经采纳了这个概念，而消费者们也越来越熟悉。值得注意的是多数（接近全部）公众在信任的基础上选择进入，而不会完全确认 LBS 的 TOS（服务条款）。

选择进入和选择退出的核心就是让个人来控制。在很大程度上，个人隐私会受到位置内容交付机制固有的控制水平的影响（也就是说，拉动或推送）[①]。因为在基于拉动的 LBS 中，个人在互动中具有更多的控制权，决定与服务供应商发起互动是个人的自愿行为，并且位置信息只提供给要求完成的交易（例如，通知个人最近的出租车的位置）。相反，在基于推送的 LBS 中，被追踪位置信息的目标个体，比如当出现在一个零售店附近时，将有可能接收到不请自来的信息/服务。

人们都有一个普遍的共识，就是利用选择进入程序来获得用户同意，然后再使用他们的历史位置。至于这种做法应该视情况而定，还是一视同仁，目前存在争议。尽管从可用性的角度看，人们可能不能接受一事一议，至于一刀切的做法，消费者则可能不知道他们的位置数据被共享这个事实。

在某些情况下，LBS 服务供应商无权决定如何实施选择进入程序。一些移动运营商会对面向消费者的 LBS 应用（例如找朋友服务）强加上严格的条件，以规避可能发生的隐私诉讼。尽管移动运营商也在尝试培育开放和创新的市场，但“在难以确定时，选择最安全的方法”似乎是移动运营商常见的信条。

① Heng Xu, Sumeet Gupta, Pan Shi,“Balancing User Privacy Concerns in the Adoption of Location-Based Services: An Empirical Analysis across Pull-Based and Push-Based Applications”网址为 https://www.ideals.illinois.edu/handle/2142/15224。

虽然事情总会变化，但是移动运营商和开发者花点时间搞清楚谁在管理隐私是值得的。

10.3 谁在管理LBS隐私

有五个主要的当事人决定如何管理 LBS 隐私（不包括政府机构和立法者）。并不是所有的成员都有平等的发言权，并且一些成员的角色会很快被彻底改变。这些成员是[①]

- 移动网络运营商
- 手机制造商
- 位置整合者
- 第三方开发者
- 互联网公司

移动运营商及其对隐私的态度

作为一个 LBS 开发商，你应该知道多数移动运营商对隐私的态度非常保守。假如你打算让自己的 LBS 应用在手机运营商那里高调上架或预备高调推出，那就得考虑额外多采取一些保护隐私的措施。

现在我们将了解隐私管理的这五类成员详细信息：

- 移动网络运营商——对移动生态系统而言，幸运的是所谓的“围墙花园”（移动运营商为控制应用发行设置的限制条件）正在土崩瓦解。直到最近，运营商还在控制整个 LBS 的价值链条。他们是唯一能够通过控制台技术访问用户位置的实体，同时最初常常只让由服务器托管的、属运营商品牌的第三方应用来访问这些信息，而屏蔽所有其他应用的 GPS 功能。如果你不是移动运营商推荐的对象，那很可能只有依靠长尾效应在市场上勉强度日。而我们发现，移动运营商正在推动所谓的“开放环境”。不仅专门针对开发人员，而且新应用上市的流程也顺畅了许多。

  同时，这并不能阻止 MNO 对访问其位置平台的用户施加一些限制。尽管前面讨论过的推送机制（自动告诉我的位置，并向我推送相关的信息）能给用户带来巨大的利益，但是除了用消费者 LBS 检测用户位置的拉动机制（用户必要要求确定其位置）之外，目前几乎没有一家网络运营商采用推送机制。
- 手机制造商——一些手机制造商，比如诺基亚，通过在兼容 Secure User Plane Location (SUPL)的手机上提供辅助 GPS 服务逐渐替代了 LBS 价值链中运营商的角色。这样，诺基亚就像是用户隐私的护卫。其最近有关对第三方开发人员开放 Ovi Maps 的声明使该公司的这种护卫作用显得更加重要。诺基亚利用用户位置资料建立 Cell ID 和 Wi-Fi 热点参考

① Claire Boonstra, Guus van Knippenbergh, Sander Meijers (Open Mobiel Internet Initiative—OMI2), “Location Based Services on Mobile Internet” November 2008, 网址为 http://sprxmobile.adix.nl/wp-content/uploads/2008/11/final-lbs-whitepaper-final-nov-2008.pdf。

数据库，从而向室内的终端用户提供可替代的定位功能。

- 位置整合者——为了公开运营商的位置资产并创造额外的利润，北美的运营商比如 Sprint 开始与位置整合者比如 uLocate、WaveMarket (现在称为 Location Labs)以及 LOC-AID 合作，通过第三方开发商访问位置资料。很多情况下，位置整合者会取代运营的护卫角色。
- 第三方开发商——SUPL 标准的问世让人们可以在任何一款 GPS 智能手机上安装任何第三方的 LBS 应用。重要的是，用户在决定加入之前，可以通过检查开发商的可信赖性来保护自己的隐私。通过控制安装哪些应用，他们就成为自己隐私的保卫者了。
- 互联网公司——随着具有地理定位功能的移动浏览器和 LBS 应用的出现，隐私控制权又落在了互联网公司的手中，比如谷歌，这些公司能提供像 Mobile Maps 这样的应用，包括当地搜寻，在活动范围内寻找朋友以及社交网络解决方案。随着手机和传统 Web 的界限日益变得模糊不清，大规模整合已经完成，互联网公司很有可能在管理移动设备隐私上扮演更重要的角色。

10.4 隐私立法

欧盟在 1995 年通过了一系列保障 LBS 用户隐私的法规（欧盟区国家都在执行），这些法规还在不断改进，以适应相关技术的变化。根据 EU 隐私条令（95/46/EC、97/66/EC 及 2002/58/EC IV），在部署 LBS 时必须要遵循三个关键原则：披露、准许和数据安全。每条原则的主要含义如下：

(1) 披露——任何具有位置数据收集职能的公司应该向消费者披露该公司要收集哪种类型的消费者数据，以及收集这种数据的目的或用途。数据收集者的透明度是该原则的关键。

(2) 准许——在收集用户个人数据之前，数据收集者应该获得数据所有人的准许。这也就是指用户位置数据使用的选择性加入和选择性退出。

(3) 数据安全——收集到的数据应该得到充分的安全保护，避免意外丢失、盗用、披露、非法处理或者类似的事情发生。

10.4.1 避开数据隐私陷阱

如下指导原则可以有效地帮助 LBS 开发者和企业家遵守有关个人数据的隐私法规，缓和用户对隐私的担忧并且避免引起诉讼：

- 首先不要收集数据。
- 如果必须收集数据，不要储存它。
- 如果你真的必须要储存数据，就要对数据作匿名处理。
- 如果你真的需要储存数据并有能力保护，就要加密这些数据。

避免法律咨询费用

如果一家公司保存了追踪个人位置的日志，那么就会受到相关法律的制约。这样的制约可

能表现为不同方式（包括非正式质询、传唤，或者担保），来自不同的团体（执法部门或诉讼代理人）。对于遵从某项请求是否是法规规定的、非强制性的，或者甚至是法律禁止的，是否有责任风险，存在着复杂的法律问题。

这种复杂性甚至可能涉及国际法。例如，当欧盟公民在美国使用美国公司的服务时，则该美国公司在欧盟的分支机构也应该遵从欧洲的数据保护法规。

拥有较大地区性数据集的公司会面临较大风险，律师以及执法部门有朝一日会知道该数据集，并开始利用法律程序得到它。避免这种昂贵法规风险的最好方法就是首先不要获取可辨认的位置数据[①]。

首先不要收集数据

这个说起来容易，做起来难，因为收集数据有时候是不可避免的。虽然限制数据保留是保护隐私很重要的方法，但最好的保护方法还是不要记录信息。开发人员有必要花时间好好想想什么信息是绝对需要收集的。如果你不能确定一些数据是否需要，最好就不要收集它。

如果你必须收集数据，不要存储它

LBS 供应商应该仅在需要的时间内保留用户的位置信息，然后就销毁这些数据或者使这些信息不可读。如果为了长期使用必须保留位置信息，那么有可能的话，LBS 供应商应该将这些位置信息转化为汇总数据（看下一点）。

因为储存空间很便宜，并且越来越便宜，现在能抵制住不储存数据的诱惑会更重要，这通常是阻力最小的一种途径。如果你不得不暂时储存数据，为了确保删除的数据确实删除了，要注意使用安全删除工具。

如果你真的必须储存数据，就要匿名处理数据

大多数 LBS 服务总在某种层次上存储数据，基于地理（住宅区、城市、国家）、种族、年龄或其他变量给个人历史信息分组。这种数据汇总可以使第三方利用该信息而又能保护匿名的个人用户。但也应该知道，即使位置数据库去掉可辨别的标签，也仍然可能泄露信息。

例如，如果你知道约翰是唯一住在布鲁克银行街的用户，那么那个在布鲁克银行街使用基于位置服务的人自然可能就是约翰。一般来讲，解决这种问题的方法就是在人口密集的区域限制使用基于位置的服务，尽管在一些情况下这种方法并不可行。

如果你真的需要储存数据并有能力保护，就要加密这些数据

利用加密和精心设计来保护位置隐私，从一开始就要考虑周到。这种方法并不廉价，并且越来越多地用于高度敏感的资料（比如财务记录）。现代密码学支持将数据处理系统设计成为遵守各种隐私法规的系统，从彻底隐姓埋名到有限的匿名以支持执法。尽管数据加密并不廉价，但 LBS 使用者和服务供应商都对它非常信赖。

现在我们已经知道了和位置数据有关的数据存储问题，接下来再了解一下目前的隐私立法支持哪些最佳实践。尽管并没有法律规定要遵守这些指导方针，但最后，随着更严格更具体的法规

① Blumberg et al.，“On Locational Privacy”.

出台，遵守这些指南是很明智的。

10.4.2　移动电话产业协会的最佳实践指南

移动电话产业协会，或 CTIA，公布了一些对 LBS 服务应该如何应对隐私法规，以及怎样在移动运营商或无线运营商以及 LBS 应用供应商之间分配责任的建议。CTIA 建议的三大基本原则是通知、准许和保障。

给提供位置数据的用户回报

移动服务的用户越来越意识到他们提供给 LBS 服务商的位置数据的价值。在广告领域，用户已经对同意在手机上接收广告信息时得到一些回报抱有期望。在欧洲和美国，LocatioNet 通过广告补贴向用户提供一种免费的导航服务。尽管目前对很多销售商而言，完全靠广告收回服务成本可能会很困难，但是他们至少应该向选择展示广告的用户提供优惠。

类似地，“用免费服务回报对位置历史的访问”越来越受欢迎。比如 TomTom MapShare 就将位置数据用于提高服务的质量。对谷歌和诺基亚公司而言，过去利用私人的位置数据建立小区 ID 和 Wi-Fi 热点参考数据库，同样适用这种情况。

用户帮你收集信息，还要一分不少地付费显然是不道德的。所有人都必须认识到：“位置宝藏”的获取是要付出代价的。

表 10-1 列出了 LBS 应用对应的不同角色的责任，明确指出了要求同意和通知的领域。

CTIA 建议 LBS 供应商应该通知用户，尤其是当位置信息用于 LBS 以外目的的时候。由于某些用户没有意识到或者无法施加控制，这就导致了隐含或明确许可的区别。比如员工监控，可以是隐含许可，也可以是根据合同规定的其他合理的许可方式。

CTIA 还声明 LBS 供应商们必须允许 LBS 用户撤回其先前向所有人或特定的第三方披露其位置信息的同意授权。在技术上可行的条件下，LBS 供应商要根据 LBS 用户或无线运营商账户持有人的要求，向用户提供选择性终止功能，或根据 LBS 用户类型限制个人的 LBS 应用（见表 10-1）。

表10-1　美国CTIA基于位置服务的最佳实践指南明确区分了无线运营商/移动运营商和LBS应用提供商之间的职责

无线运营商	应用提供商
无线运营商也是LBS供应商，因为它向第三方提供位置信息。	应用提供商是LBS供应商，因为它能从无线运营商那接收位置信息从而向移动用户提供LBS服务。
无线运营商应该通知其账户持有人： ❑ 该设备支持定位； ❑ 经过授权的用户可以随时随地使用主界面或主菜单中的位置应用； ❑ 启用服务后，账户持有人即认可在使用 LBS 时向第三方公开用户的位置；	应用供应商应该通知LBS用户： ❑ 为了提供服务要收集用户的位置信息； ❑ （将会/将不会）把位置信息透露给他人； ❑ 只有在提供服务需要的时间内保留位置信息（例如向地图上的 LBS 用户提供其最近的 ATM 位置信息）；

（续）

无线运营商	应用提供商
❑ 它有可能会保留用户的位置信息，并且只要有业务需求就会使用 LBS； ❑ 用户应该反复阅读隐私政策从而明白该应用是如何使用及保护位置信息的； ❑ 如果不愿与第三方应用供应商分享其位置信息，用户就不应该启用该服务；见第 4.A 节的例子。	❑ 可能会通过移除或模糊处理可辨认的个人信息来产生汇总的位置信息； ❑ 汇总的位置信息可能被用于提供基于位置的广告； ❑ 不再给出其他通知或提醒。
通过购买具有定位功能的无线服务，账户持有者同意无线运营商向第三方应用供应商披露用户的位置信息。	用户同意该服务的条款和条件。

关于保障，CTIA 给出下列位置信息安全的建议：

> LBS 供应商应该使用合理的管理、物理以及/或技术手段，以保护用户位置信息不会遭到未经授权的访问、修改、破坏、使用或披露。LBS 供应商应该在合同约定情形出现时适当保护用户位置信息的安全、完整和隐私。

最后，它建议 LBS 供应商应该向用户提供报告滥用的资源，并能提供及时解决数据滥用的程序。

无线运营商为用户提供的移动设备能够访问由第三方软件支持的地图服务。该无线运营商向第三方提供用户的位置信息，相应地，要通知该地区服务的用户。

隐私国际，一个倡导隐私法的民间组织

隐私国际（PI，privacy International）是 1990 年成立的一个人权组织，它作为一个监督机构来监控政府和公司侵犯隐私的行为。隐私国际总部设在英国伦敦，在华盛顿特区设有办事处。它在世界各地开展隐私宣传运动，目的在于保护人们的隐私不受政府和公司的侵害，以防隐私权遭到侵蚀。它相信隐私是自由的根基，其目的是利用一切办法保护隐私。目前，隐私国际不向具有良好记录的公司提供任何形式的认证，这是一个有待满足的需求。

本节最后，我们将了解一下 LBS 生态系统中的移动公司如何满足法规要求，同时把对位置感知应用性能的影响降到最小。

10.5　遵守隐私法规

既要搞定隐私恐惧，又要遵守隐私法规。作为开发者，有哪些选择呢？

- ❑ 设置用户资料
- ❑ 选择加入屏幕
- ❑ 不确切的位置信息
- ❑ 服务条款
- ❑ 地理围栏

下面我们分别介绍这些工具，适当的时候也会给出实际的应用示例。

10.5.1 设置用户资料

很多 LBS 提供商在应用中包含了设置和特色功能，以便让用户管理和控制他们的隐私①。例如，Google Latitude 让用户选择对其好友可见或不可见。可以调整的设置没有限制，比如包括：

- 谁看到什么（让同事能看到我在办公室干什么，但看不到我在其他地方干什么）
- 什么地方是私人场合而什么地方是公共场合（我家的位置总是私人的）
- 他人是否可以根据距离远近联系或添加用户（我不想在公共列表中出现）
- 最近的位置如何处理（我想在 24 小时后再共享我最后所在的位置）

最终，需要在充分地隐私保护和用户体验之间找到平衡。设置也应该是灵活的，而且对用户也尽量是友好的。尤其是用户应该能够很容易地在任何时候关闭位置共享功能。

类似地，设置功能要定义什么时候以及哪些位置能被共享，让终端用户感到一切都尽在掌握。手工设置会降低用户体验，很多用户经常会忘记打开或配置应用。有些 LBS 应用让终端用户完全控制，只允许手工设置位置共享；用户可以输入地址或在地图上点击自己的位置来决定在什么时候以及在哪里来共享位置。

诺基亚尝试着将灵活设置与便捷实用结合起来，在匹配当前位置与预定义的喜爱地点基础上，让用户有选择但自动地来共享位置。只有当用户位于或接近一个公开定义的地点时才会共享位置，而用户不需要采取任何操作。

花样翻新或更智能的常规设置方法（还要考虑在不同服务商之间导入导出心仪地点的需求）为将来用户设置管理指明了方向。

10.5.2 选择加入屏幕

10.2 节介绍了选择加入对遵守 LBS 的基本隐私规则是必要的。实践中，应该把发布选择加入提醒常态化。至于怎么实现，可能会由于移动运营商和使用的移动开发平台而有差别。例如，在运行 Symbian OS 的诺基亚设备上，可能会对每一次追踪到的位置发送选择加入信息。在 iPhone 上，更为典型的是使用一次性的选择加入屏幕而不是不断重复的提醒。图 10-4 展示了 Starbucks 和 AccuWeather 应用中选择加入的 iPhone 截屏。只有用户选择加入，这两种应用才能向用户发送有意思的结果，因此在现实中，如果用户想使用该应用那么要选择退出的机会就很小。

有些人说选择加入功能有一个最大的问题，即缺乏向用户告知其位置数据使用频率及目的的信息。在 Google Maps 上线时，用户并不知道自己要为谷歌公司创建小区 ID 和 Wi-Fi 热点参考数据库做贡献，而该数据库能作为室内不能使用 GPS 时的定位依据。

① Dominique Bonte, “Exploiting the Location Goldmine While Respecting Privacy—A Delicate Balance,”ABI Research.

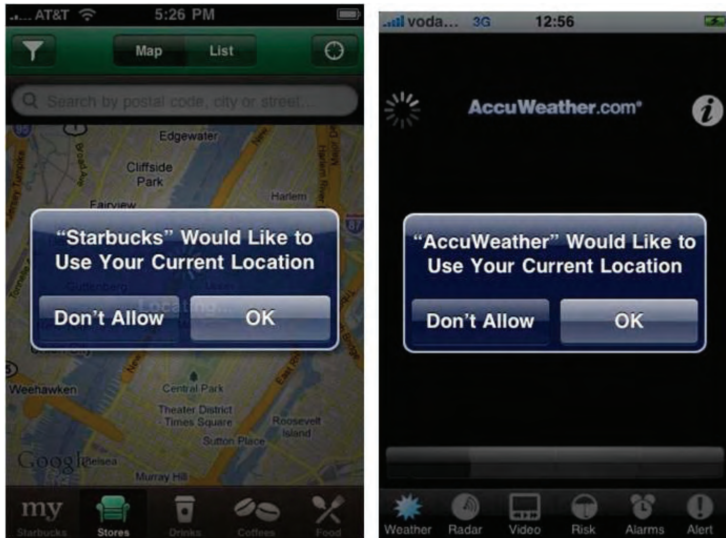


图10-4　Starbucks和AccuWeather应用中的选择加入屏幕

很明显，当涉及社交网络应用时选择加入是一个更敏感的话题，尤其是这些应用是对公众开放的。The Dopplr iPhone 应用就是与用户设置有关联的一个很好的选择加入的例子，在用户位置（选择加入）基础上，该应用能让用户寻找周围相关的信息，但同时又能使其位置痕迹保持为私人状态（选择退出）。相关的 Dopplr 应用截屏如图 10-5 所示。

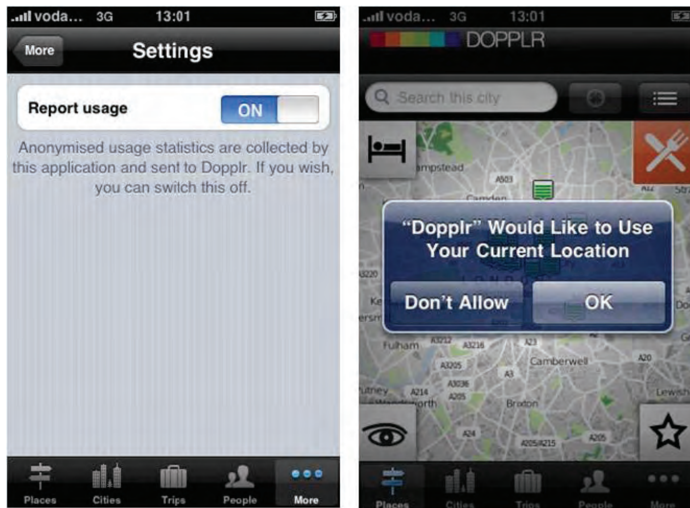


图10-5　Dopplr iPhone应用的截屏，用户可以同意使用他们目前的位置来浏览附近的服务或人员，但选择退出等于不授权应用访问其私人位置信息，不能用于匿名统计

选择加入屏幕对保护隐私很重要，但是另外一种保护 LBS 应用隐私的办法就是使用“不确切的”位置信息而不是提供确切的信息，这将会在下面介绍。

10.5.3 不确切的位置

保护隐私最受欢迎的方式就是共享不确切的位置而不是有 GPS 坐标的确切位置。分享不准确的位置曾经并且仍然是不具有 GPS 功能手机上常用的唯一方法。尽管以 Cell ID 和 Wi-Fi 为基础的替代定位技术正变得越来越普遍，但是这些方法并不能提供与 GPS 定位方法相同的精度。降低精度也是 GPS 手机一种特意保护隐私的措施，只分享附近的或城市的位置而不是确切的坐标。同时，降低位置信息的空间精确度——但也是暂时的——既限制了用户体验也限制了向第三方提供历史位置数据的有用性。

10.5.4 服务条款

一个好的服务条款（TOS，Terms of Service）协议是任何位置感知应用的必要部分，一谈到隐私保护，多加小心点是没有错的。例如，Centrl（一家美国的 LBS 供应商）推出了一种安全政策，将最后知道的用户位置信息保留一周，而并不储存用户的位置信息。如果用户在一周以上的时间里没有登录，其最后的位置信息也会被删除。（可以参考 GyPSii 的完整 TOS，将其作为一个起点：http://corporate.gypsii.com/content/view/8/）。GyPSii 也像 Dopplr 那样匿名处理用户行为数据，像其他 LBS 服务一样将数据合并存储。

10.5.5 地理围栏

地理围栏（geofencing）是位置感知应用领域中相对较新的一种成果。地理围栏是真实世界地理区域的一个虚拟边界。其基本概念就是让用户在其附近或其他位置绘制虚拟的围栏，在这里用户能让位置服务知道他们在哪里，以及他们不喜欢哪些地方。这样，根据用户是否处于围栏之内，就可以触发某种行为。

开发者可以设置一种动态地理围栏（比如，围绕一个特定位置的半径范围）或一种静态的地理围栏（如学校周围或住宅周围）。

从隐私角度来说有个很有意思的想法，就是让用户设置自己的信息封锁区域，在这里活跃的移动应用将无法获悉用户的位置。另外，地理围栏可用于触发推送通知（10.2 节我们已经介绍过了）。当用户进入某个特定区域时，就能自动地在 Foursquare 这样的服务中签到。

本章最后，我们了解了一些位置感知应用在移动设备上使用时遵守隐私法规的一些实用指南。尽管要涵盖所有潜在的个人隐私问题很困难，但是我们介绍的这些工具则都能确保很好地遵守隐私法规。

10.6　小结

隐私仍是位置感知或基于位置的服务中备受争议的话题，人们对隐私的观点出现了严重的两极分化。一些人认为在线隐私不存在，而一些人则希望自己对私人生活能全部控制（或许没有想到真正的隐私在当今的数字时代只是个乌托邦而已）。这种争论很有可能在一段时间内会继续存在，而服务供应商应该明智地遵守法规并促进数据使用的透明化。越来越多地，如果应用公司能够证明其应用有可靠的隐私保护，说服消费者相信他们提供的服务会更稳定、隐私保护措施会更可靠，那么他们将会有很大的竞争优势。随着我们的时代继续向实时在线、实时更新的方向迈进，处理更多更复杂隐私数据的挑战也会增加。成功的基于位置的服务将需要努力权衡，既要保持应用简约，同时又要让用户始终掌控应用所能展现的内容、对象和时间。

随着竞争越发激烈，推出的应用程序越来越好，让用户很容易地发现你的明星应用至关重要。下一章，我们学习应用分销的主要内容，从而确保你的付出有个最好的回报。

第 11 章

应用分销

本章内容

- ❑ 选择正确的分销策略
- ❑ 发布应用
- ❑ 让他人发现你的应用
- ❑ 通过第三方分销

上一章，我们介绍了与位置感知应用相关的隐私争论，以及怎样确保应用遵守隐私指南和法规。

在本章中，我们将会了解如何使应用在日益增多的应用中脱颖而出。内容包括如何为你的应用选择正确的分销策略，如何发布应用，以及可以使用哪些第三方分销渠道来分销应用。

11.1 产品好不好，要看卖得好不好

目前，全世界移动应用数量已经超过 50 万种，来自 8 个开发平台，为确保目标用户注意到你的应用，分销显得前所未有地重要。简单来说，好的分销策略能使你的应用成为万众瞩目的焦点，而不好的分销策略会埋没你的应用。

尽管开发一种伟大应用的气魄很值得钦佩，但事实上伟大的应用至少有几千种之多。好的消息是移动用户总是希望使用新应用，而且有一种尝鲜的冲动。确保用户可以发现你的应用，你的成功概率会大大提高。

在决定如何发布应用以及选择哪个分销渠道之前，你需要回答一些有关应用开发目标的基本问题。例如，如果你开发的应用只从美国高速公路（但没有其他地方）的摄像头中取影像，那么就没必要选择全球性分销平台；使用美国国内的就好了。

你的应用寿命短吗（比如跟下个月要上映的电影或举办的体育赛事紧密相关）？那么就不要指望在移动运营商的门户上发行；因为光签订门户发行协议可能就要花 12 个月时间。

本节中，我们将讨论开发者需要思考哪些问题，从而做出最好的分销决定；我们会用一个决策树模型来帮你勾画出不同的结果。

分销平台决策树模型

确定了应用的目标市场（第 5 章为你提供了相关信息），使用一个简单的决策树（如图 11-1），你就可以找到大致的方向。很明显，这种模型也会有意简化许多变量。因此声明是少不了的。不只因为“App Planet”（GSMA 的说法）或者应用商店始终在不断变化，而且应用商店之间的区别也越来越少了。不过，此模型能让你对自己的应用找到大致的方向。开发者可用的主要分销平台有四种：

- 独立的应用商店
- 运营商支撑的分销渠道
- 手持机制造商（OEM）的应用商店
- 操作系统（OS）商店

为了简化模型，我们把 OEM 和 OS 商店放在一起，但本章其他部分要分别对它们进行介绍。另外，运营商支撑的分销可以采用门户在线（即移动运营商在其门户上预先加载该应用，这样就不用下载了）或者运营商应用商店分销的方式。

选择正确分销策略的技巧

选择正确的策略，让世界知道你创造性的应用，需要在进入费时、费钱的阶段前，深思熟虑一番。

- 你需要回答 5 个基本问题：
- 你的应用是为谁设计的？
- 你计划推向哪个地域市场？
- 应用的发布时间重要吗？
- 你的应用是与运营商、操作系统，或者手机型号绑定的吗？
- 你的应用是免费的吗？

这个模型包含四个关键决策，每一个都能导致不同的结果：

- 决策#1：应用价格——如果应用需要对下载它的用户收费，就需要一个支付平台。领先的独立应用商店 GetJar 还不支持收费，因此对收费应用来说不合适。OEM 和 OS 应用商店能为应用提供最好的支付机制（尽管利润分享方式不同）。在实践中，Android Market 和 iTunes Store 和 GetJar 一样，都是很好（甚至更好）的免费应用分销渠道。独立应用商店更适合那些以功能手机用户为目标产品及市场上的长尾产品。
- 决策#2：期望的上市时间——随着基于模板的开发工具（如 Golden Gekko 的 Tino DIY 移动应用开发工具）越来越多，开发时间正在减少。这使得一次性的、短期的用完即丢的应用越来越多。这些一次性的 LBS 应用中，很多可能是简单的定位应用。例如，为了发布一部电影，可以开发一个显示首映影院位置的应用。对这类时间要求很高的应用，没必要忍受移动运营商长期的批准程序，整个过程可以轻易地花掉你 12 个月的时间。另一方面，独立应用商店通常 24 小时就能批准程序，并能在要求的时间内上架。

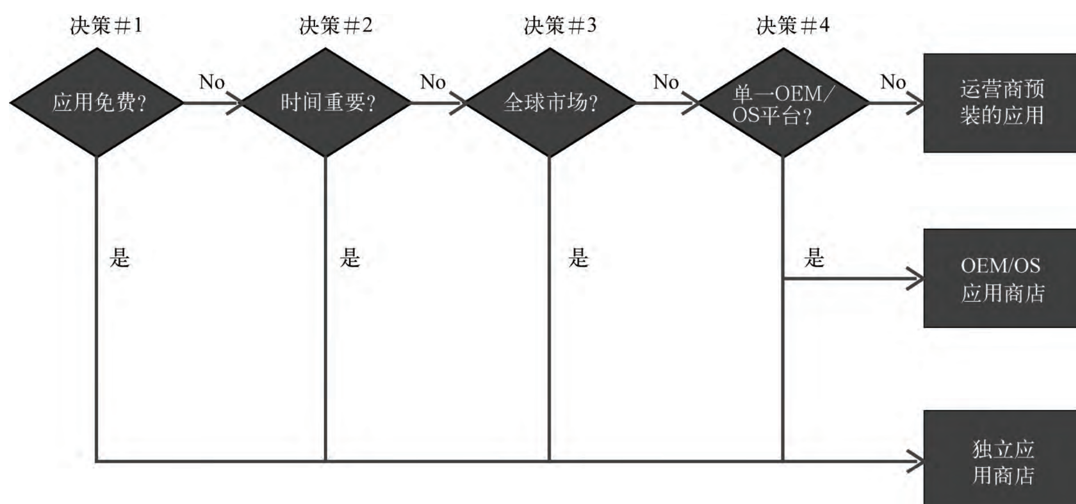


图11-1 根据四个关键应用标准基础上选择最佳分销平台的决策树

- 决策#3：地理目标市场——尽管一些应用平台实际上面向全球，但运营商还是会区别国家不同对待。获得了全球移动运营商在某个国家（比如英国）的批准，不一定能够获得在另一个国家（比如德国）的批准。这使运营商很适合本地应用分销。如果你的应用针对全球市场，那么支持多平台、多 OS、多设备型号是更好的选择。此时，你可以同时在独立应用商店和单一 OS 或 OEM 商店中分销应用，从而确保其能真正地覆盖全球。
  如果你的目标是全球市场，也要注意不同的独立应用商店会有不同的用户组成。例如，Mobile9 在亚洲有大量的用户（尤其在菲律宾、马来西亚和印度尼西亚），但欧洲的用户就很少。
- 决策#4：OS/OEM 平台——由于手机市场高度分散，这会是一项很难做出的决定，多 OS 和 OEM 是理想的默认选择。在实践中，对于较小的开发商来说，开发基于多种平台的成本实在太大，因而被迫选择一个热门平台。随着智能手机的普及（现在智能手机的销售增长最快），iPhone 和 Android 设备已成为热门平台。

在新兴市场国家，标准手机仍有较高的市场占有率（有时超过 90%）。多数这样的手机只能运行 JME/Symbian 应用。因此地域的选择可能会决定 OS/OEM 平台的选择。只专注一个 OEM 商店？或许不行。

不是所有的利润分享都是平等的

不同的分销商有不同的利润分享协议。尽管应用商店的利润分享标准固定在 30/70（即 30% 的佣金支付给应用商店，70% 支付给开发者），而移动运营商协议会根据应用类型、运营商本身、分销的国家及其他很多因素而不同。在多数情况下，运营商最多支付 50/50 的利润分成（即只有 50% 付给开发者），但运营商佣金高达 70% 也不是没有过。

11.2　发布应用

现在你已经很清楚怎样制定正确的分销策略了，接下来我们将了解当今的主要应用商店，包括独立的应用程序商店和链接到个别制造商或运营商的应用商店，以及如何选择正确的应用商店中发布应用。鉴于 iTunes App Store 为所有应用商店做出了表率，本节最后会给出在该商店中发布应用的例子。

11.2.1　应用商店概况

今天，开发者可以选择的移动应用商店至少有 58 个①（并且还在快速增长！），我们来看看怎么从中选择吧。

移动内容整合者的注意事项

应用商店变得非常流行，而媒体也不断宣传新应用商店推出的消息。在许多方面，鉴于下载应用正日益成为一种冲动购买行为，这种情况也是合乎情理的。

大量的传统应用仍要通过移动内容整合者分销。移动内容整合者是专门致力于移动内容（传统媒体、铃声、壁纸、游戏或奇特的程序）整合的公司，通常会与许多移动运营商签署协议，为其移动门户网站提供移动内容。

尽管通常情况下，移动内容整合者确实只会整合传统的移动内容，但他们同样也能为基于位置的应用整合内容。他们很可能会制定严格的选择标准，因为其权限将在很大程度上取决于移动运营商本身的要求。这些条件对知名企业的已通过 12 个月左右测试的成熟程序将更有利。

作为主要的全球移动内容整合商，Zed 与世界各地超过 130 个的移动运营商都有合作协议。

正如我们前面所说的，移动应用商店有四个主要的类型：

- 独立的商店（例如 GetJar，如图 11-2 所示）
- 手机制造商（OEM）商店（例如，BlackBerry App World，如图 11-3 所示）
- 移动运营商商店（例如，BlueVia）
- 操作系统（OS）商店（例如，Android Market）

① App Store Catalog, WipConnector, 2010 年 3 月, http://www.wipconnector.com/appstores.

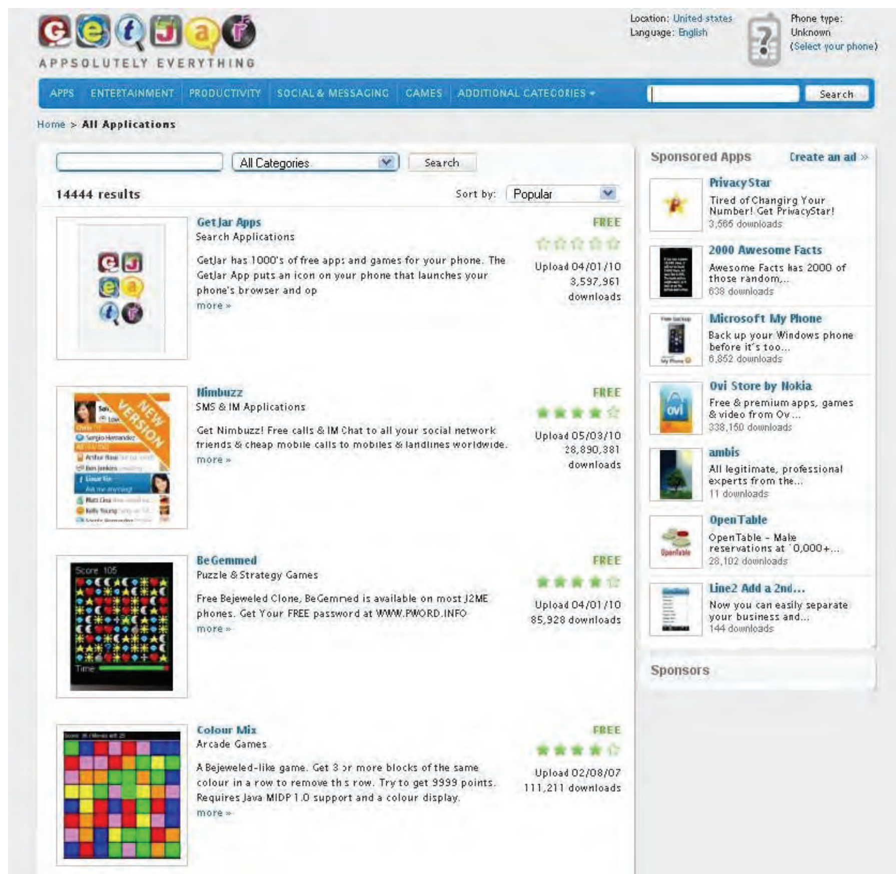


图11-2 GetJar是独立的全球性应用商店，拥有适用于所有移动OS（除了LiMo）超过5万种应用，但目前仅提供免费的应用

分类很简单，但是每种商店都有自己的目标，这一点一定要搞清楚：

- 独立商店将自己定位为应用开发的创新先锋；他们鼓励小的应用开发者以尽可能少的花费、尽可能短的审查时间在线。越来越多地，独立商店寻求在我们第 8 章讨论过的长尾市场中为特定的群体提供新的产品（例如，MiKandi 只以成人为目标用户）。通常独立商店从大流量中赚取利润，也就是广告分成（也有一些专业商店收取佣金）。
- 手机制造商商店，比如 iTunes App Store，努力推销自己手机或操作系统，通过销售应用来赚取利润（Apple 收取 30%的佣金）。

  他们喜欢大力促销在其手机上而非其他手机上安装的应用程序，并主动（免费）宣传他们认为的面向新领域、有市场吸引力的应用。诺基亚等手机制造商最初推出应用商店的目的，在于促进应用市场繁荣，从而销售其手机。随着越来越多的制造商从硬件制造转为软件开发，他们的目标也发生了变化，现在他们更关注从应用商店中赚取利润。



图11-3　BlackBerry的App World拥有超过3000种应用，可以在世界各地20个以上的国家使用，只收取应用开发者零售20%的佣金

手机制造商商店的优势一般体现在对各种机型的支持上，另外下载过程也简单。但也不尽然，Nokia 的 Ovi Store 就是一个特例。它在克服初期的各种困难（从应用认证到复杂的应用支付）后，取得了多方面成功。

- 移动运营商商店（迄今为止，仅有几家）是在 iTunes App Store 获得成功之后作为补救推出的，意在延续其在移动应用世界的控制权。此类型的例子如图 11-4 所示。



图11-4　移动运营商商店，比如Vodafone的 Betavine和O_2的Litmus，主要是试验性的，目的是在将来在大众市场推出应用商店，比如Telefonica的 mstore

他们并没有打算震撼自己的客户，Vodafone (Betavine)和 O_2 (Litmus)最初进入这一领域更多是试验性的。现在他们又重新推出了 Vodafone 的 App Shop（前身是 Vodafone 360）和 Telefonica 的新 mstore。

- 操作系统（OS）商店是由 Android 运动（如图 11-5）导致的相对较新的产物，但是现在被其他厂家所复制。不同商店之间的界限正变得模糊，韩国三星正计划推出自己的智能手机平台 bada，以及一个独立的应用商店。

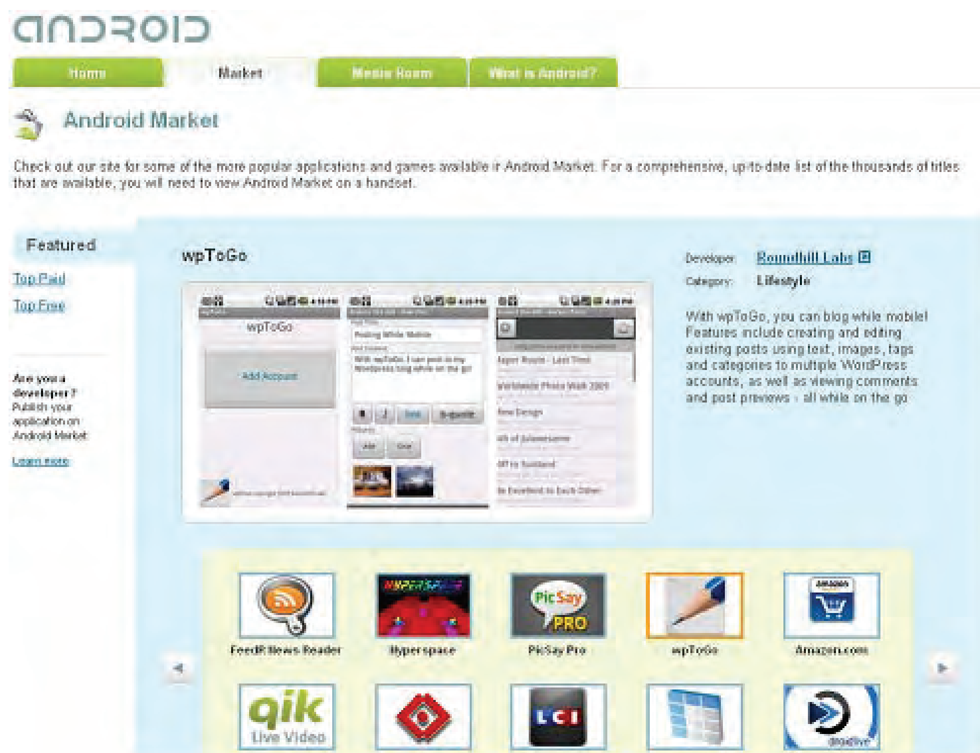


图11-5　The Android Market OS商店提供的应用数量较少但是增长迅速

图 11-6 在一个矩阵中画出了全球比较流行的应用商店，根据这些商店面向本地的或全球，以及是否仅支持一种 OS 平台进行了分类。每个应用商店的圆圈的大小表示应用数量，因此有着 35 万种应用程序和超过 100 亿下载量的 iTunes App Store，圆圈比其他的都要大些。而且这还只是为 Mac OS 和一种设备的封闭环境开发的应用。尽管 Android Market 仍相对较小，其开放的开发环境鼓励了越来越多的开发者向这里靠拢，据预测，在一年或两年之内，它将很快赶上 iTunes App Store。

第 6 章表 6-3 有不同应用商店的统计信息。要了解所有应用商店，WIP Connector 有定期更新，网址是 www.wipconnector.com。

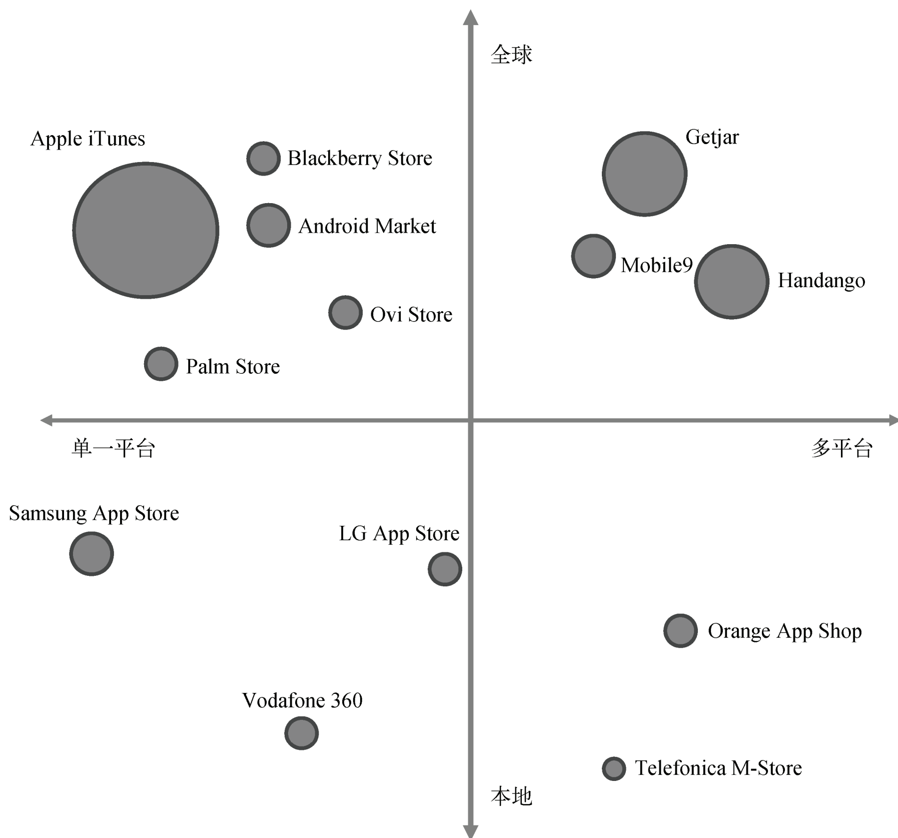


图11-6　主要应用商店的位置，包括优势明显的单一平台全球商店（iTunes遥遥领先）和多种平台的全球商店（通常是独立的）。圆圈的大小代表了商店中的应用数量

11.2.2　怎样在商店中发布应用

那么，在一个主要的应用程序商店中发布应用需要哪些步骤呢？你到底需要做什么？我们将举个最流行的应用商店 iTunes App Store 的例子，模拟与其有关的发布过程。在开始之前一定记住三个关键的先决条件：

- 一张信用卡
- 一部 iPhone 手机
- 一台 Apple Mac PC

然后按表 11-1[①]列出的步骤做。第 3 步和第 4 步在本节正文中都有详细介绍。

① iPhone Distribution Build Cheatsheet, iPhoneDeveloperTips.com, http://iphonedevelopertips.com/wpcontent/uploads/2009/iPhoneDistributionBuildCheatsheet.pdf.

表11-1　在iTunes App Store发布iPhone应用的四个主要步骤

步　骤	做　法
1	加入如图11-7所示的Apple iPhone Developer Program 重要提示：要加入Apple iPhone Developer Program（如图11-7所示），你需要一个iTunes账户，且必须支付99美元的费用才能在商店中上传应用
2	确保你有最新的Apple开发工具 Apple向开发人员提供大量优秀的开发工具，从而帮助他们完成开发。确保已经下载了最新的Apple SDK以及Xcode iPhone开发环境（包括一个图形调试、源代码编辑器和项目管理工具）
3	创建iPhone应用 重要提示：在构建应用之前，你需要安装一个发布配置文件。另外，要完成下面这些步骤， Xcode首选项中的General选项卡必须设置为All-in-one布局 A. 创建部署配置 B. 设置目标信息 C. 设置活动配置 D. 验证目标设置 E. 验证nfo.plist
4	添加应用 注意：这样做，你将需要： A. 一个应用名称 B. 一份应用功能的说明（用你所希望支持的多种语言） C. 最多5张应用截屏，指定其中一张放在应用商店的首页 D. 该应用所属的主要类别（如旅游）和次要类别（如娱乐） E. 一系列的搜索关键词，通过这些关键词在iTunes App Store中就能找到你的应用 F. 应用的支持URL G. 应用的URL

第 1 步：加入苹果 iPhone 开发者计划

这个步骤在表 11-1 中描述过了。

第 2 步：确保你有最新的开发工具

看表 11-1 中的说明。

第 3 步：构建 iPhone 应用

A. 创建部署配置。

(1) 在 Xcode 中选择项目名称（如图 11-8）。

(2) 点击右键，选择 Get Info Dialog（或按 Command-I）。

(3) 选择 Configuration 选项卡。

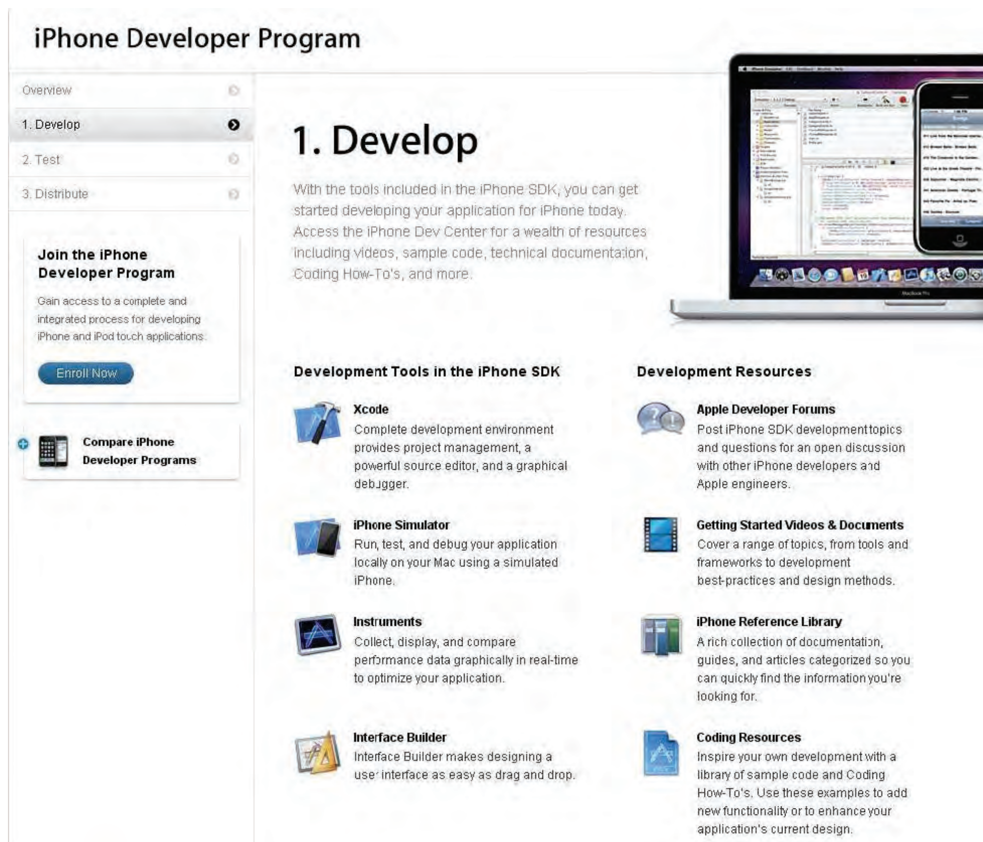


图11-7　Apple的iPhone Developer Program提供很多开发工具，包括Xcode和Apple Instruments，还提供Apple应用开发最佳实践的逐步指导

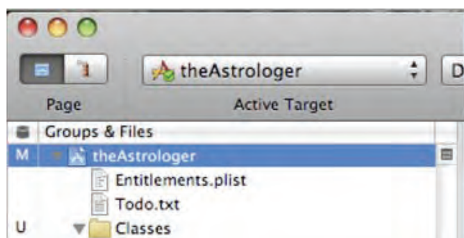


图11-8　使用Xcode需要先给项目命名

(4) 在配置列表中点击 Release，在底部选项中选择 Duplicate。

(5) 给新的配置部署命名。

(6) 关闭窗口。

B. 设置目标信息。

(1) 选择目标（看图 11-9）。
(2) 选择 Build 选项卡。

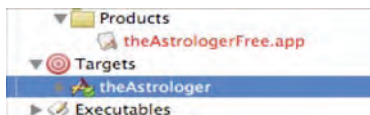


图11-9　给你开发的项目命名后，需要在选择SDK之前选择目标

(3) 点击右键选择 Get Info Dialog。
(4) 从 Configuration 下拉列表中选择 Distribution。
(5) 在设置列表中 Architectures 部分选择一个 Base SDK（例如，Device - iPhone OS 2.2.1）。
(6) 在 Code Signing Identity 下面的 Code Signing 部分，选择适当的配置文件（例如，Ad Hoc 或 Distribution Profile）。
(7) 选择 Properties 选项卡。
(8) 设置 Executable Name（例如，theAstrologerFree）。这里有个默认值，${EXECUTABLE_NAME}，也适用于多数项目。
(9) 把 Identifier 设置为 com.domain.application-name（例如，com.3SixtySoftware.theAstro-loger-Free）。其默认值为 com.yourcompany.${PRODUCT_NAME:identifier}，可能适用于你的项目。如果出错或与其他应用冲突，试试把${PRODUCT_NAME:identifier}替换为其他唯 一字符串，可以代表应用即可。
(10) 把 Icon File 设置为 Icon.png。
(11) 确定使用 57 x 57 像素的图标，名称为 Icon.png。
(12) 设置 Version #（例如，1.1）。
(13) 关闭窗口。

C. 设置活动配置

(1) 点击 Debug 工作区按钮（见图 11-10）。

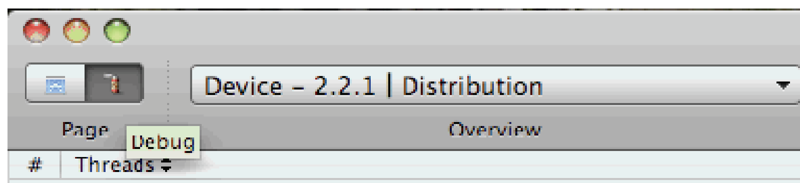


图11-10　通过访问Debug工作区，并选择Distribution as the Active Configuration

(2) 从（左上角）下拉列表中选择 Device 设置，即 Active SDK 对应的目标（例如，Device-iPhone OS 2.2.1）。
(3) 选择 Distribution as the Active Configuration。
(4) 点击 Project 工作区按钮（该按钮在 Debug 的左边）。

如果你要创建一个临时的配置，按下面的步骤：

(1) 创建一个新的文件（Command-N），选择 Code Signing，再选中 Entitlements，点击 Next。

(2) 将文件命名为 Entitlements.plist。

(3) 取消选择 Get-Task-Allow 复选框。

(4) 保存文件。

(5) 确保该文件位于项目层次的根目录下（也就是说，拖放该文件到该项目名称下）。

(6) 选择 Target。

(7) 点击右键，选择 Get Info Dialog。

(8) 选择 Build 选项卡。

(9) 用 Entitlements.plist 填写 Code Signing Entitlements。

D. 验证目标设置。

(1) 选择 Target。

(2) 点击右键选择 Get Info Dialog。

(3) 选择 Properties 选项卡。

(4) 看一眼 Executable 名称（例如，theAstrologerFree）。

(5) 选择 Build 选项卡。

(6) 向下混动到 Packaging。

(7) 验证（或输入）Product Name 与上面的 Executable Name 匹配。

(8) 关闭窗口。

E. 验证 Info.plist。

(1) 在 Resources 文件夹中点击 Info.plist。

(2) 检查下面内容：

—Bundle Display Name：这是将在 iPhone Home 屏幕上的图标下显示的文本。

—Executable Name：这应该与你在 Properties 选项卡中输出的名称匹配。

—Icon：必须设置为 Icon.png。

—Bundle Identifier：例如，com.3SixtySoftware.theAstrologerFree。

—Bundle Version：例如，1.1。

F. 清理和构建

(1) 在 Build 菜单中选择 Clean All Targets。

(2) 在 Build 菜单中选择 Build（Command-B）。

第 4 步：添加应用

这是非常简单的一个步骤，如果你手头上有所有信息应该用不了 30 分钟。你必须有下面的信息，以便在主菜单中选择 Add New Application 提交（看图 11-11 和图 11-12）。

- 应用名称
- 应用功能的描述（用你希望支持的所有语种）
- 至多 5 张应用截屏，从中指定一张作为在应用商店中展示的主页面

- 一个主要类别（例如旅游）和一个次要类别（例如娱乐）
- 一系列的搜索关键词，通过这些关键词就能在 iTunes App Store 中找到你的应用程序
- 应用的支持 URL
- 应用的 URL

Add New Application

Overview

The following information will appear with your application in the App Store.
Required fields are in **bold**.

Please provide this information in **English**.

Application Name :

Application Description :

Do you want to restrict this binary to a specific platform? :

Yes No

Primary Category : Select One

Secondary Category : Select One

Copyright :

Version Number :

SKU Number :

Keywords :

图11-11 在iTunes开发者区的主菜单中选择Add New Application，你可以看到一系列屏幕，其中为一系列必填的信息，包括应用的完整描述以及应用种类等

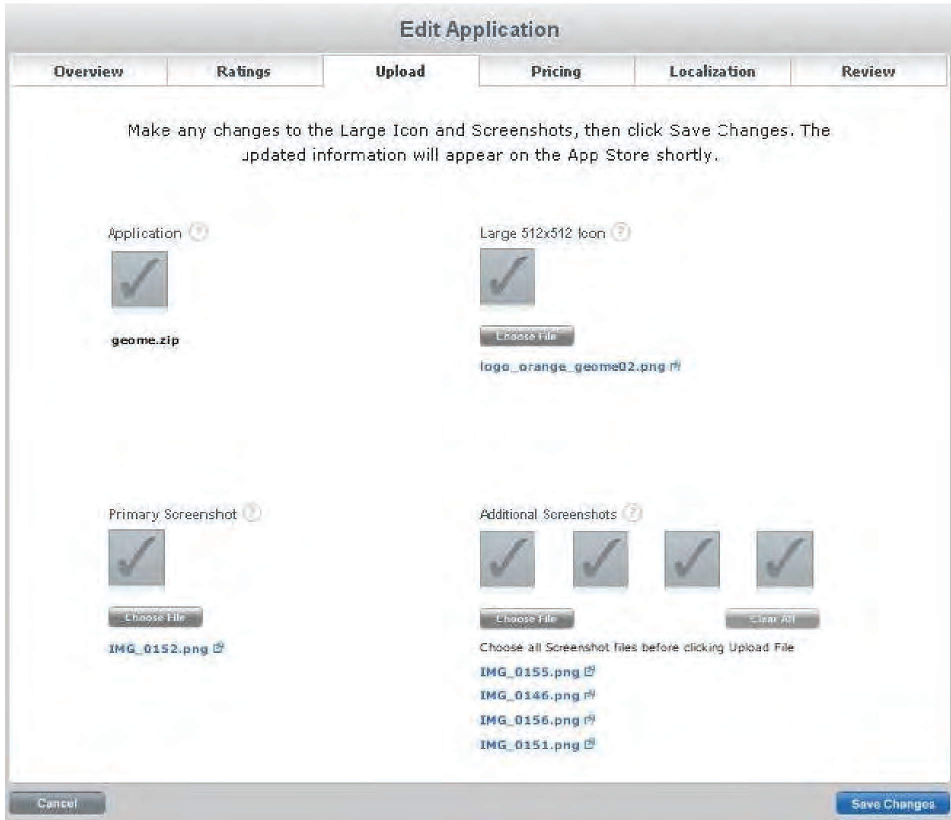


图11-12　在iTunes开发者区输入应用名称和描述后，还要为应用添加一个商标，至多5张截屏

恭喜！你已经在应用商店中发布了第一个应用!通过我们介绍过的四个关键步骤，你已经上传了所有信息，然后你就可以等待 Apple 批准该应用了。对于遵循 Apple 指南的应用来说，通常批准需要 2 到 5 个工作日。

现在我们应该制订计划,以确保你刚刚发布的应用能受到目标群体的关注。我们将通过应用一些可发现性技巧，实现这一目标。

11.3　应用商店的可发现性

人们对应用商店到底是如何运作还不甚了解，很多经营者也不想向公众透露。即使是应用商店的基本信息，比如商店中可用的应用总数，有时也向公众隐瞒。尽管有报道称 Apple 的 iTunes 商店的下载量可能会达到 30 亿次，人们同时也在质疑有多少应用下载后又被移除了。

首先，坏消息是：让消费者在一个应用商店中发现或者找到你的应用很难。iTunes App Store 就是一个典型的例子：在 30 多个国家的独立商店中，从超过 14 万种的可用应用中找出并不知名

的应用是相当困难的。多数用户是通过浏览其 iPhones 手机上排名前 25 位，或者最受欢迎的应用类别发现应用的。不这样做的人很少。

现在，好消息是：通过了解应用商店是如何运作的，你就可以显著地提高应用流行的概率。为避免在这个琢磨不透的世界中迷失，开发者应该时刻铭记自己的首要目标：让更多人发现、查看并下载你的应用。一个好的便于发现的策略将考虑下列内容：

- 应用评论
- 应用排名
- 应用分析
- 应用可发现性服务

接下来，我们介绍这些内容的详细信息，以便你掌握它们并取得最好的效果。

11.3.1　应用评论

谈到评论，似乎很明显，必须先有好产品，然后才能有好评。也不用说，你的应用应该在上架之前就经过了充分测试。应用一旦获得差评是很难扭转局势的。

但你应该知道提交评论的过程不一定是公平的。只有将应用从其设备中移除时，用户才会收到提示，请他们提交评论。而想删除，往往意味着你不喜欢该应用。因此许多评论更倾向于负面反馈也就不难理解了。

怎样才能获得好评呢？专家承认一款能提供某种独特功能的应用，再加上一个完美的用户界面，就能得到用户的好评，而千篇一律的应用就不可能赢得善变用户的欢心。

评论能影响排名吗？答案是否定的——或至少并不直接影响。排名是由下载量，而不是满意度决定的。

11.3.2　应用排名

那么，应用排名到底意味着什么？名次又是怎么决定的呢？排名是指一个较短的时间内被下载应用最多的应用，通常是 24 小时。列表中通常是排名前 10，前 25 或前 100 位的应用。因为排名定期更新，应用在列表中的位次会上下变动，但要排进该列表则需要相当大的努力。

一些应用商店，像 iTunes 和 Android Market,会公布自己的排名，可以用 App Gems（如图 11-13）或 App Rank 工具监测全球排名前 300 的 iPhone 应用。

为什么排名很重要呢？简单来说，排名越靠前，应用的下载量越大，因为会有更多的人接触你的应用。获得排名对应用的成功来说至关重要，这是让用户大量下载应用的唯一方法。这种正相关的关系可以参照图 11-14，该图展示了 TripIt iPhone 应用排名与其获得下载数量的关系。

根据领先的移动分析工具 Pinch/Flurry 统计，在 iTunes App Store 排名前 100 的列表中，每天新增的用户数量能达到 2.3 倍。正如你所预期的，如果排名前 10 或前 25，那么每天应用新增的用户数量会更高。要想排进前 25 位，你需要有每天 2 万次的下载量，要是降到每天 5000 次，就会排到前 100。至于应用使用情况，仅有总下载程序 1%的应用能够天天使用，用户使用收费应

用的时间要长于免费应用。



图11-13　iPhone应用App Gems的截屏，用户可以通过它监测其他iPhone应用的全球排名情况

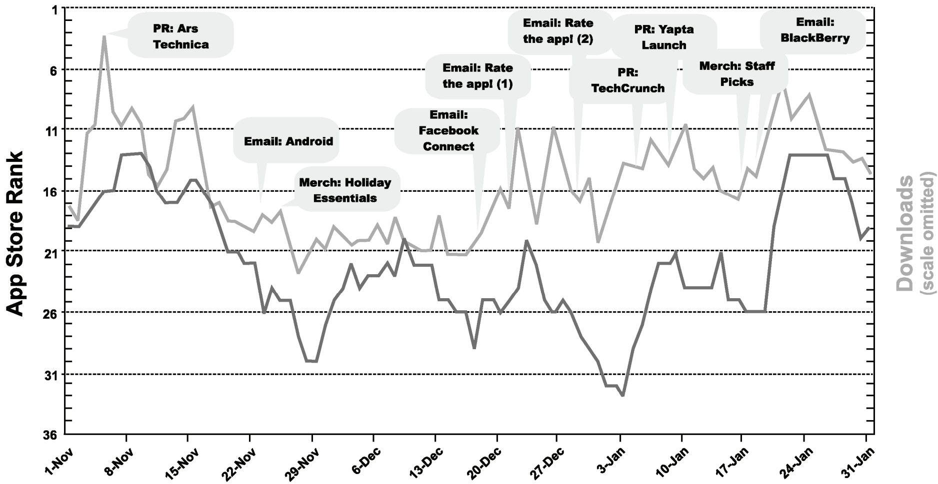


图11-14　TripIt在iPhone商店排名及下载量，可见排名与下载量之间有较强的相关关系。下方的折线是从11月1日到1月31日三个月内的排名情况。上方的折线展示了同时期该应用的下载量。随着排名的下降，比如11月22日的一周，应用的下载量也跟着下降。随着排名上升，比如1月17日的一周，应用程序的下载量也跟着上涨[①]

① Developer Secrets: Increasing App Store Sales; San Francisco, February 8, 2010 presentation, Will Aldrich,TripIt, mng.bz/q144.

怎样使排名靠前呢？有个秘诀，涉及三个元素：

- 发布前推广——在应用推出前开始为其造势。给有可能评论的人发邮件，看他们能否对你即将推出的应用给出评论。
- 发布计划——计划多种发布方案。在第一次发布时，不要提供全部功能。吊起用户的胃口，声明在将来某个时候发布所有功能。然后定期发布新版本从而推动应用的销售。
- 持续营销——成功的营销是关键。对很多受欢迎的应用来说，广告结合病毒式营销是很有效的方法。在本章后面我们会介绍病毒式营销。

谈到广告，你工具箱中的关键工具就是交叉应用广告。它通常表现为横幅广告的形式，点击下载选项，就把用户带到应用商店相关的版面。

交叉应用广告商必须考虑到，多数应用的生命周期都是很短的（因此短广告爆发会比渗透性广告更好；参照图 11-15，Mixology 应用的短广告爆发在排名上的成效），并且也必须权衡广告的成本/收益（也就是，将每一次下载的投入与收入做比较）。

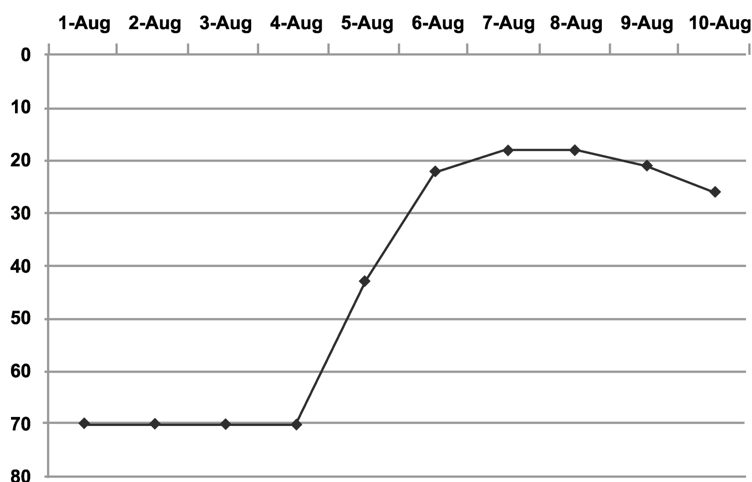


图11-15　Mixology iPhone应用中简短移动广告爆发对排名的影响，展示了推出活动两天之内急剧的上升趋势（来源：AdMob）

应用分析主要集中于测定应用的下载量（通过市场、设备等），但是也越来越多地考量营销活动对排名和下载量的影响。

11.3.3　应用分析

正如你在本章开头看到的，一个关键决定因素就是应用是否免费。事实很简单，总体上，免费应用的下载量更大，排名更靠前。如果你的应用是免费的，就会希望用户不断地过来下载（有许多会话），并尽可能长久地使用它。这是获得应用印象和广告收入的驱动力（也是对免费应用

的支撑）。使用像 Flurry Media 这样有效的监测工具可以监视访问人次和会话时间。Flurry Media 提供一种仪表盘式的监测工具（如图 11-16 所示），它能完整地统计访问人次、用户的地理位置，并且对于多平台的应用，能分别统计应用在不同设备上的使用情况。

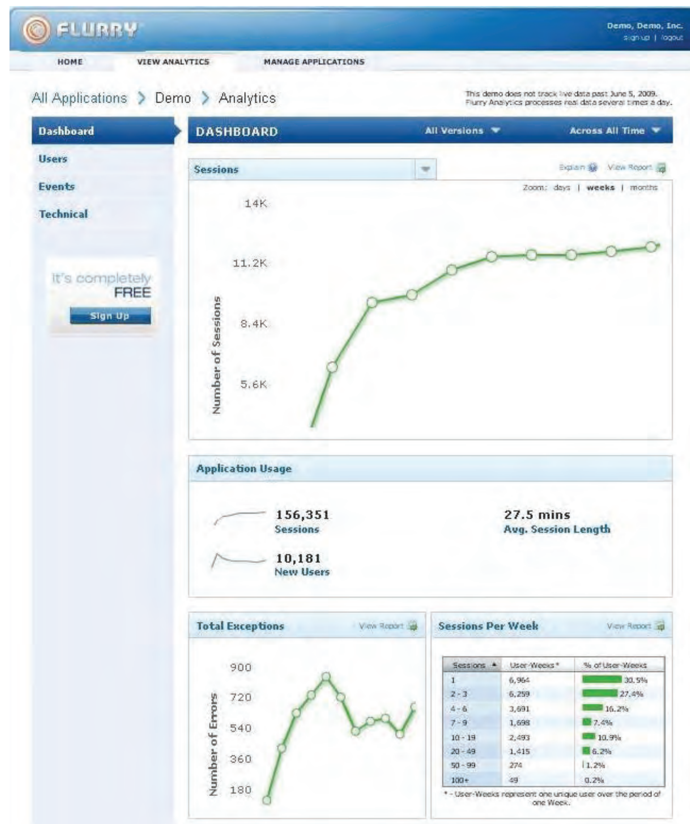


图11-16 Flurry Media提供一种仪表盘式的免费分析工具，让开发人员追踪用户的会话以及会话持续的时间，从而评估该应用是否达成了其目标

因为有很多试错在里面，这些统计结果对打造应用分销策略来说非常有帮助。应用价格定在 4.99 欧元还是 2.99 欧元好呢？降低/提高价格，看看会发生什么。

11.3.4 应用可发现性

刚刚介绍了交叉应用广告对下载量的影响。还有其他免费的方法可以让用户发现你的应用，甚至不需要应用商店中的排名或评论。最近，冒出大量帮助消费者寻找应用的网站，它们要优于传统的（有时令人困惑的）个人应用商店，可以分为下面几类：[1]

① iPhone App Store Statistics, Mjelly, 9th November 2009 http://blog.mjelly.com/2009/11/iphone-app-storestatistics.html.

- 应用整合网站，像 Apptism、AppShopper 和 148Apps
- 应用评论网站，像 AppVee、AppStoreHQ 和 AppCraver
- 应用社会化推荐网站，像 Appsfire、Yappler (图 11-17)和 AppBoy

应用整合网站会对已有的应用列表再发布，但是会提供一个不同的更友好的用户界面。应用评论网站包括专家对个人应用程序的评论，目的是不断地筛选出最好的应用。

应用社会化推荐网站有许多附加价值，能告诉用户他们的朋友都在使用什么应用，当这些网站与流行的社交网络结合后，就可以起到很大的口碑作用（我们将会在下一节中讨论）。Yappler 的用户界面是同类网站中最友好的（如图 11-17 所示），而且支持在 Twitter 和 Facebook 上以及通过电子邮件来分享 iPhone 应用。



图11-17　Yappler典型的应用社会化推荐网站，可以让用户向其朋友推荐自己喜欢的应用，最后对应用价格做实用的整合统计

最后，值得牢记的是最先在第 8 章中提过的自由增值赢利策略。你知道应用货币化的流行策略就是先提供免费版，然后再提供一个溢价版。这就意味着免费版的应用可以激起用户使用溢价版应用的兴趣（例如，在用户使用 10 次后自动弹出通知）。

11.4　通过第三方分销

和前面讲的直接分销技巧一样，开发者（及其公司）也能寻找第三方推动应用的交叉销售，并通过他们的渠道刺激分销。现在我们会了解一下内容合作伙伴和 WOM（口碑，Word-of-mouth）渠道。

11.4.1　通过内容合作者分销

通过内容合作者分销应用以增加该应用曝光机会的方法常被开发者忽略，因为这确实需要开发人员额外做一些准备工作（以及外出搜集情况的工作！）。

但是，到底什么是内容合作者呢？一个内容合作者是指不直接参与移动应用开发，但有其自己专属媒体的公司。

它可以是

- ❑ 照片分享网站（像 Panoramio）
- ❑ 点评网站（像 Yelp）
- ❑ 社交/媒体网站（像电影社区 Rotten Tomatoes）

在很多情况下，可以与合作者协商，相互提供交叉销售的机会。因此，在广告战中这种方法可以让开发者花很少的钱（或者不花钱）就能够实现成效卓著的效果。

内容合作者和移动应用开发者双方互惠互利，就能长期合作。Flixster 电影应用和 Rotten Tomatoes 影迷社区网站就建立了一种有效的合作关系，后来 Flixster（如图 11-18 所示）成为 iPhone 应用中领先的电影应用。



图11-18　Flixster与Rotten Tomatoes的有效合作使Flixster在广告战中花很少的钱就能在iPhone应用中获得额外的曝光机会

旅游应用 TripIt 也与很多旅游网站有合作关系，包括票价追踪网站 Yapta。这种合作推动 TripIt 发展成为旅游应用的领导者。

11.4.2 通过口碑销售

如果你运气好，能有大笔的营销预算，那么传统的广告就是提高应用知名度的很好方法。但即使在这样的情况下，维持应用的知名度也需要口碑宣传或社会推荐来传播。

应用的口碑最终取决于应用到底是否能满足目标用户的需要，这是口碑的基础。任何应用推出时都应该在营销计划中考虑三个因素。

社交媒体

在线社交网站是推广应用最有力（以及划算的）的方法之一，它能有效地创造口口相传的广告效应。市场细分的结果就具备了某些社交网络（脸谱网和其他）、博客（像 TechCrunch）以及其他的在线功能。有效利用这些功能的杠杆效应，不仅能够得到目标用户的支持，还能使他们成为该应用的市场推广人员，从而进一步扩大用户范围。

一个好的社交媒体策略能识别有效促进应用传播的人群或网站，而且要包含下面的因素[①]，：

- 病毒性营销——这有赖于消费者得到应用的好处并传播它（就像“传应用福音的人”）。好的适合病毒营销的应用能很容易“分享好消息”（比如一键启动短信息或邮件推荐）
- 热门话题营销——通过消费者对应用的谈论来宣传。这种方式适合能为市场带来真正全新的或时事话题的应用。你也可以利用已有的热门话题（比如社交网络宣传）。
- 权威营销——让意见领袖传播应用（通常要支付一定费用），从而使你把握言论导向，如果权威人士有很多追随者，就会有直接的社会影响。

交易会

现在的贸易活动有各式各样的形式和价位，以取悦不同口味的消费者。从展览会（世界三大手机大会是 Mobile World Congress、the International Consumer Electronics Show 和 CTIA），到研讨会、专业网络会议，以及初创企业比赛（比如 Red Herring 100）。

对于靠自身努力的初创企业而言，创新者竞赛是一种有效地应用促销方法。O' Reilly 组织的 Where 2.0 是美国位置感知初创企业的主要大会。而在欧洲，LeWeb Paris 是初创企业的主要竞赛活动（虽然这个活动是为初创企业举办的）。但是要警惕，竞争是相当激烈的，不是所有的创新者竞赛都能产生同样的媒体影响。

公关行动

如果你有很明确的市场定位，具有某些媒体特性或网站功能的多功能新闻发布和公关也会非常有效。许多公关机构专门从事高科技初创企业咨询业务，对于那些有前途的公司，他们偶尔会接受该公司的股票而不是现金。在竞争激烈的 IT 领域，有专家为你做公关传播可以给你带来极大的价值。良好的公关会使你的应用从籍籍无名变成媒体宠儿。很多公关公司都专注于特定的新领域，比如技术、初创企业或者移动领域。Communications 和 Ink Communications 就是两个有代表性的公关公司。

① “The Insiders Guide to Word of Mouth Marketing” White Paper, DotMobi Resource Centre, 2009 http://mobithinking.com/word-of-mouth-marketing.

另外，推荐看看这篇文章：*Word of Mouth Marketing: How Smart Companies Get People Talking*，作者是 Andy Sernovitz（Kaplan, 2009），你可以了解此领域的更多细节。

11.5　小结

好的应用能创造新闻。

iPhone 让用户在移动设备中下载有趣的应用变得极其方便。消费者们希望得到好的产品，并且准备像买棒棒糖一样冲动地购买这样的应用。在展示出的很多好应用之中，胜出的是那些更容易被发现的（就像收银台旁边的棒棒糖）或在消费者印象中极好的应用。

本章介绍了为你的应用选择正确商店、利用分析工具试销应用，并采用营销技巧让世界发现应用的方法。随着应用商店的数量快速增加，消费者可选的应用数量也呈指数级增长，开发者如果想取得成功，就要立志做出应用中的超级明星。

下一章也就是最后一章，我们会介绍位置感知和 LBS 创业公司取得成功的最后支柱：坚守你的商业理想。

第 12 章

坚守你的商业理想

12

本章内容

- 为公司作战略规划
- 为实现战略寻找资金
- 坚守经营策略

人常说移动领域的一年堪比其他经济领域的五年。在快速变化的环境下，开发者企业家需要利用所有可以支配的工具来规划、适应和发展业务。周密的计划、到位的资金以及对知识产权的恰当保护，将使你更容易获得成功。

如果你按照本书的内容做了，那么你可能已经构思好了自己的位置应用，而且也发布并开始卖了。本章将带你走进下一步，即在应用的基础上建立并开展业务。我们将了解怎样在团队或企业核心价值的基础之上制订商业计划。然后在讨论如何确立并坚守经营理念之前，怎样正确融资以实现你的商业目标。

12.1 战略规划

战略规划包括具体的行动计划及长期目标，也可以包含公司理念、执行人、实现方式、地点及时间等。

不管怎样，战略规划的本质就是帮你做出决策。好的战略规划可以让你做出明智的决定，因为它需要在今天的决定中考虑明天的需要，例如：

- 怎样销售应用或服务？
- 向谁销售？
- 如何战胜竞争对手？

这些问题需要大量的时间思考。一旦有了答案，你就可以将其编制到商业计划中。它将成为你开始创业的一张路线图。但是在开始编制计划前，你需要选择一个正确的经营策略。

12.1.1 选择正确的经营策略

生活充满了选择。创业和经营也一样，可能会有无数种选择，但能利用的资源却是有限的。

在确定好你将要开发和推出的服务种类后，你就需要制定一个可持续发展的商业模式，尽力发挥企业的核心优势。

首先要思考企业核心价值何在。核心价值就是独特的技能和文化，是你公司的核心。对照核心价值能帮助你决定应该专注哪种产品、哪个程序或哪类消费群体。在图 12-1 中可以看到这个模型。

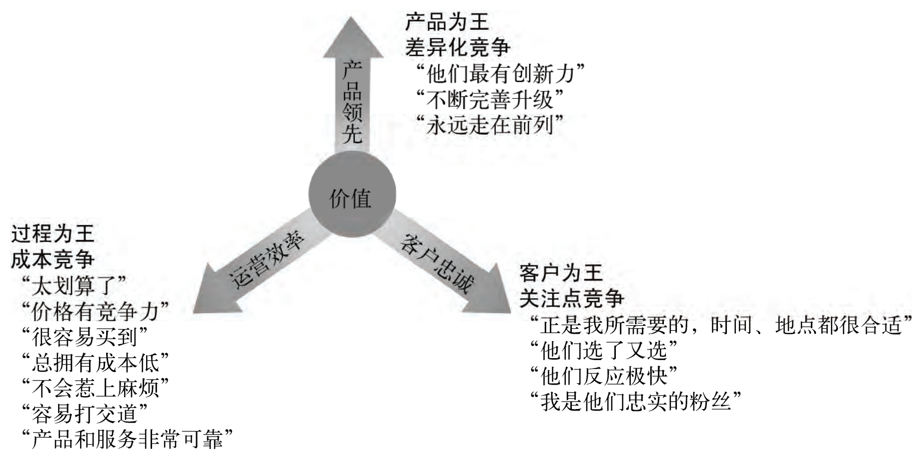


图 12-1　从公司核心价值得出的有效创业策略包括产品领先、卓越经营以及客户关系[①]

产品领先

产品领先就是在市场中制造前沿创新性产品。Apple 的 iPhone 就是领先产品的一个例子，该公司的核心价值就是传播知识。另一个例子，Android 的应用开发商 Ecorio 推出碳补偿（carbon-offsetting）LBS 应用，成为早期的领先产品，如图 12-2 所示。这使他们在 2008 年获得了由谷歌公司颁发的 Android Developer Challenge（机器人开发者挑战）奖。就像很多初创企业一样，尽管开始取得了成功，Ecorio 后来的发展很艰难。12.1.3 节将会介绍一些发展策略。

卓越经营

卓越经营就是要成为性价比最高的生产者，能为客户的花费提供巨大的价值。韩国三星与其竞争者相比，比如摩托罗拉，就有着高效的生产过程，因此在过去的 3 到 5 年中快速地获得了市场份额。作为手机应用开发商，其中一种低成本基础就是要依靠波兰、乌克兰或俄罗斯等新兴经济体中具有较强技术实力的软件开发团队。

客户关系

这就是说在与公司的每一次接触中，将客户放在第一位。客户满意度是采用这种策略的公司的服务核心。Zappos 的电子商店是一个在线公司的例子，其首页如图 12-3 所示，它有着清晰的目标：将客户放在首位。与 Amazon 的 30 天退货期限相比，顾客在 Zappos 中购买的商品可以在

① 来源: White paper by Mark A. Zawacki, “Startup Candy Vol. 1,” The Milestone Group, November 2009。

365 天之内退货（或许 Amazon 在 2009 年 11 月完成的对 Zappos 的收购，也证明了 Zappos 独特的以客户为中心的文化是一个决定性因素）[①]。

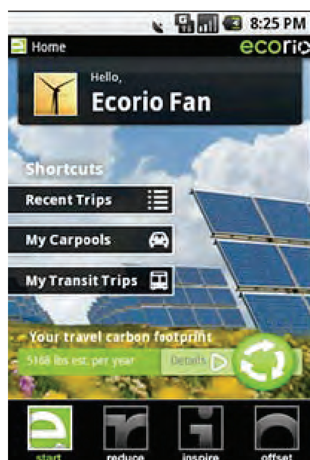


图 12-2　Ecorio 在 2008 年 8 月推出绿色 Android LBS 应用，在得到 Android Developer Challenge（机器人开发者挑战）奖之后，建立了自己早期产品领先地位。此应用能帮助用户减少二氧化碳的排放



图12-3　Zappos是一个著名的案例，该公司从一开始就制定了明确的“顾客至上”的公司策略，以期获得在线客户的青睐

思考过核心价值之后，接着就要制定一个可以取胜的经营策略，使其与你的商业计划吻合。

① Sarah Lacy, “Amazon-Zappos: Not the Usual Silicon Valley M&A,” *Business Week*, July 30, 2009; available at http://www.businessweek.com/technology/content/jul2009/tc20090730_169311.htm.

12.1.2 制订营销策略：商业计划

从某种意义上讲，多数企业家都需要制订商业计划。这是一个 10 到 100 页的文件，描述了公司的产品、雇员、营销以及财务计划。它说明了你的（新）产品目标、为什么这个目标要优于其他目标，以及怎样为投资人赚钱。商业计划的写作有较高的要求，它迫使企业家考虑可能会影响其商业理念的所有关键内容。

制订商业计划有时候也是一种考验，因为它要求许多思考和分析（可能导致分析麻痹），需要很长的时间（在某些情况下，需要几个月）。不论你是否撰写商业计划，都强烈推荐你思考商业计划的内容。这可能让你有一些意外收获，能为你提供建设性的意见，从而帮助你完善想法。

作为写作计划的一部分，你可能首先要向外界披露你的经营理念，因此在这样做时有几点值得注意：

- ❑ 大多数人（假设他们都是诚实的，而你要成为一个企业家）将会有一大堆的理由说你的想法行不通。对于这种情况，你需要充分相信自己的理念。
- ❑ 有人会告诉你早就有和你相似的想法了。你需要准备好解释为什么你的想法是不同的/更好的。
- ❑ 有人会详尽地问些怎样实现想法。要准备好自由分享此项目的所有方面（除了“神奇配方”，如果你有的话）。

最好不要把商业计划看成是一成不变的。我们生活在一个高度竞争的时代。你需要适应它。你可能在今天还没有竞争者，但是明天就有了，因此就需要调整计划。关键是尽你所能把商业计划写得简短（不管怎样，有些人只会看摘要！）。你应该用 10 到 15 页充分描述可能遇到的风险。如果需要的话，以后可以再加上。在某些情况下，你可能需要彻底重写该计划的大部分内容，丢掉先前的版本，所以说简洁是必要的。

尝试将计划分成几个关键的部分，这样就能在变化时快速地更新。基本上要包含下面内容：

- ❑ 竞争者概述
- ❑ 市场预测
- ❑ 财务预测
- ❑ 商业设想

附录 B 给出了一个典型的商业计划大纲。

最后，要记住商业计划既是一种思考，也是沟通。你要构思和提炼自己的想法以便清楚准确地解释经营理念，并用它说服将来的客户和利益相关者。记住，该计划不可能反映现实，但是只要你已经考虑了可能出现的不同局面，即使环境改变，它也能提供一些指导。

编制商业计划的要和不要

要

- ❑ 自由分享你的经营理念和商业计划，尤其对将来的客户。
- ❑ 准备好解释为什么你的想法与现在的不同。

- ❑ 学着解释你的想法，即使是你的母亲也能听明白。
- ❑ 记录建设性的批评。把你的商业计划看成一个可以改变的文件。
- ❑ 把你的计划当作一种活的资源或可变的文件；常常更新。

不要

- ❑ 过分保护自己的想法。
- ❑ 受制于评论人负面的“不能做”说法。
- ❑ 希望你的计划有无限的适用期。它会终止得比你想的快很多。
- ❑ 被你的计划所束缚。它不会反应现实。

例如，假设你在计划中声明将要在第一季度的广告宣传中花费 5 万美元，但实际上花了 15 万美元。只要你有好的理由解释为什么这样做（例如，结果创造了 50 万美元的收入）或者有个预料之外的商业计划情况需要大笔的广告投入，那么就没有问题。不要被死的商业计划所束缚。

12.1.3 发展策略：外部合作伙伴

如果要考虑公司的整个生命周期，从创建到倒闭，存在一个“漏斗效应”。在 100 个新建的公司中，可能平均只有 25%到 30%的公司能度过前三年。其中一个原因就是与公司的发展有关，特别是要正确平衡可持续增长的问题。这种发展既不能太快也不能太慢。发展太快会造成可分配资源太少，发展太慢就会被竞争者赶上。

我们在前面提到了 Ecorio，尽管它在 2008 年赢得了 Google Android Developer Challenge 奖，但从那时起发展就停滞了，也没有抓住机会。其中一个原因是该业务是一群大学生的项目，他们有技能但却没有资源去拓展想法。对资源有限的初创企业来说，最好的发展策略是发展外部合作伙伴（或特殊的关系）。

外部合作伙伴能有多种不同的形式，比如：

- ❑ 融资伙伴（比如资金安排，不久我们就会涉及）
- ❑ 代理伙伴（让一个公关公司来宣传你的公司）
- ❑ 许可（通过第三方使产品商业化）
- ❑ 交叉推广协议（两个公司之间进行交互营销）
- ❑ 与公共机构合作（通过研究拨款或免费使用大学的科技资源）

这些不同类型的协议可以让企业家利用有限的资源，像一个更大的公司一样运作。他们也能帮助初创企业建立自己的信誉，使他们获得比自己单打独斗更为广泛的公众曝光度。

在签订这些协议之前，值得考虑的是该合作是否会让双方受益。一个互惠互利的协议比一个片面的协议更可能产生好的结果。在签订具有法律约束力的协议之前，通常要由双方签署备忘录。在这里你需要写出该协议的目标。加上这些，通常要签订一份保密协议，目的在与保护双方不受竞争的威胁。除非你真正信任与你做生意的那一方，否则应该知道此保密协议是没有意义的，你不可能依靠这个协议的保护来避免不道德的商业行为。

12.2　寻找投资

写作商业计划是确立商业理念的一个巨大飞跃。如果你已经写完了商业计划，那么恭喜你！通常，将计划变为现实，你需要资金支持。下面我们将看看四个主要的资金来源，并了解如何找到能配合商业计划的资助。在了解资金来源之前，我们将简单地考虑什么样的投资可以配合业务的发展阶段。

12.2.1　匹配业务发展阶段的投资

对一个初创企业而言，资金来源主要有四种：

- 引导资金
- “朋友、家人和傻瓜”（FFF，或者“3F”）资金
- 商业天使（也被称为非正式投资者或个人投资者）投资
- 风险投资（VC）

选择合适的投资以匹配你公司的发展阶段是很明智的。如果你仍处于设想和建立核心团队阶段，那么寻求 1000 万美元的投资没有多大意义。类似地，如果你早就收到了 VC 的资助，就不需要录求天使投资了。另外，还应该知道投资人如何划分你公司的成长阶段：

- 概念——除了一个好主意外，其他几乎什么也没有。
- 种子——你已开始定义自己的商业模式并开发产品。
- 启动——你已组建了团队并开始运营/销售。
- 成长——你已扩展了新产品线及/或市场。
- 成熟——你已达到成长的顶峰，必须通过购买其他业务来发展。

图 12-4 展示了不同类型和规模的资金怎样匹配每个投资阶段（概念、种子、启动、成长和成熟）。

现在我们已经知道了投资怎样与公司所处的发展阶段相匹配。接下来，从引导资金开始依次了解每个资金的来源。

如何识别你的理想投资者

和约会一样，如果你希望与潜在的合作者有进展，就需要做些准备。花时间考虑谁是你的理想投资者，你希望他们有什么样的素质，这样在遇到他们时就能分辨出来。

理想情况下，你希望投资者

- 有广泛的与之前初创企业合作的经验，有你所涉足的目标市场的丰富知识
- 有其他与你的业务相辅相成的投资
- 通过协作的方式，帮助提高公司投资的吸引力并建立业务。
- 具有很强的业务发展网络，能填补你的商业计划空白（比如帮助早期的企业建立管理团队、董事会和顾问团队）
- 愿意为创办人投入时间、技能和金钱

- ❑ 在投资社区（本地及国际）中与基金网络的投资合作者关系良好，有可能为你吸引共同投资人。
- ❑ 有可靠的领先记录或者有过投资交易
- ❑ 有成功创建业务并退出的历史，能为你的业务发展提供强有力的网络支持



图12-4 一个公司会根据其所处的五个不同的发展阶段：概念、种子、启动、成长和成熟，需要不同数量和类型的资金

12.2.2 引导性资助

这是指在还没有任何人投资的情况下的投资。通常，这种资助来自销售收入，但也可能来自企业创办人的其他收入。引导性资金规模通常在 5000 美元到 10000 美元之间，甚至有 20000 美元。公司的运营资本（顾客支付给你的收入与为他们提供服务所用的花费之间的差额）常由企业创办人的银行透支或信用卡暂时充抵。比如戴尔和微软公司，最初就是采用这种方式，也是当今比较常用的一种方式。而 Ross Perot 在创建 EDS（Electronic Data Systems）公司时，仅 1000 美元，但后来却将其发展成数十亿美元的企业。

引导式资助的优点就是你可以独立发展业务，不会受到外部干涉。不利的方面就是相比有额外外部资助的企业，你可能需要花费更长的时间发展业务。也可能会被不同方向的不同客户分散注意力。

如果你决定采用引导方式发展业务，就需要有与资金充足的业务不同的行为。这就意味着你需要：[①]

① Adapted from Amar Bhide, "Bootstrap Finance: The Art of Start-ups," Harvard Business Review 70, no. 6 (November 1,1992): 109.

- 快速运作
- 找到快速保本赚钱的项目
- 提供一种高价值的产品
- 不为你的企业雇佣昂贵的“好团队”
- 发展不要超过你的承受力
- 关注现金而不是市场份额
- 从一开始就要了解银行经理

12.2.3　朋友、家庭和傻瓜资助

如果你的商业想法需要比引导性资助更多的资金，可以考虑“朋友、家庭和傻瓜”（即 3F）资助。3F 资助的资金范围在 1 万美元到 10 万美元之间。这种资助方式可以让你发展一个雏形，然后就要找到更大的投资者寻求重金资助从而全面发展你的业务。如果你的业务没有计划中发展得那样好，就得做好失去朋友的准备了，找时间让每个人都了解公司的发展情况。

如果你想获得更大的投资，就要知道帕累托法则，或者 80/20 法则，它通常适用于：你的几个利益相关者可能需要花费你 80%的时间，尽管这些人只出了 20%的资金。

12.2.4　天使投资人

天使投资人（也称作非正式投资者或个人投资者）是将自己的资金投资在高潜能的初创企业上，以换取一定公司股份的个人，他也会贡献自己在特定领域的商业知识及其个人的网络资源。天使投资人通常会投资 5 万美元到 30 万美元。作为商业周期的早期、种子阶段以及/或者发展阶段的非正式风险投资者，天使投资人发挥的作用很重要。很多天使投资人都是成功的企业家，他们通常会卖掉自己的企业，有兴趣和资金来资助类似的企业。

尽管从理论上讲你可以通过正式网络和协会（例如，European Business Angel Network，即 EBAN，能将超过 25 家的天使投资网络和天使投资人聚在一起）联系到天使投资人，但事实上，没有引介者的帮助你不太可能获得他们的资助。引介者可以是天使投资人的朋友或与其有商业接触的人，他们愿意并能够将你介绍给值得信任的天使投资人。这就是为什么成功的企业家往往人脉通天，这样他们就能在正确的地方找到正确的人。

12.2.5　风险投资家

风险投资家，即 VC，是能从私人、富人和机构中获得私人自有资本的金融机构。著名的风险投资家包括红杉资本（Sequoia Capital，也是谷歌公司早期的投资者）和凯鹏华盈（Kleiner Perkins Caufield &Byers）。多数投资都集中在美国加利福尼亚的硅谷。从图 12-5 中可以清楚地看到，该图显示了从 1970 年到 2008 年美国的风险投资地图。通过联系美国风险投资协会（National Venture Capital Association），也可以找到包含 400 多个会员创投公司的完整清单。

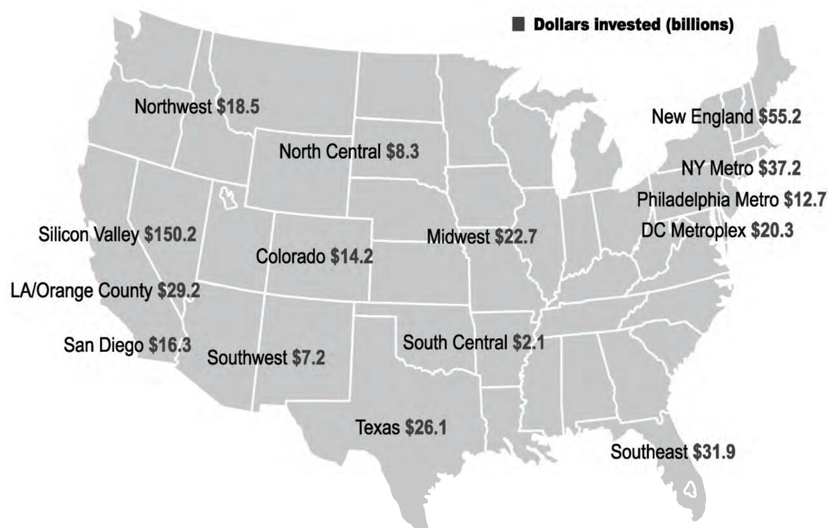


图 12-5 从 1970 年到 2008 年，美国的风险投资地图（单位是 10 亿美元，按州），该图显示加利福尼亚的硅谷投资集中，新英格兰居第二位[①]

风险投资对初创企业而言要么能助其梦想成真，要么就是一场噩梦。如果进展顺利，你的风险投资公司就会向你的企业注入大量资金，从而推动你的公司成为商业巨星。实际上，在通常情况下风险投资公司的投资不会低于 100 万美元，有时能达到 200 万美元，甚至更多。如果进展不好，你就会发现自己花费了大量的时间和精力来讨好风险投资公司，但却得不到一分钱。

风险投资公司专门做高风险的投资。这使他们对初创企业而言表现得很特别。你可以从图 12-6 中看到风险投资公司是怎样与企业家和投资银行家一起融入总体资金链条的。

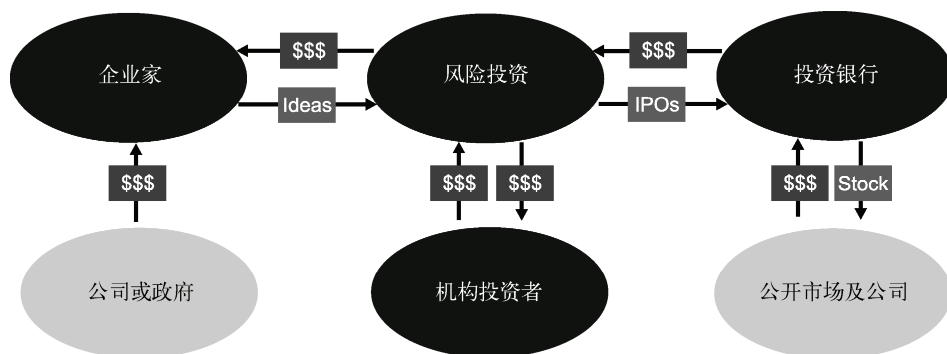


图 12-6 风险投资是怎样运作的：企业家需要钱来创办企业，机构投资者想要高回报，投资银行需要向社会出售公司。风险投资为上述三者创造市场

① 来源: mng.bz/bn45。

风险投资公司只有利用投资银行卖掉初创企业的股票才能赚钱。吹毛求疵地说，风险投资公司的目标与其公司创立者的目标并不一致，这就会形成影响他们关系的内在压力。公司创立者更希望获得成功，而风险投资公司是寻求低概率成功[①]。一个风险投资公司会将资金投资给 10 个初创企业，预计其中的 7 个会失败，2 个能撑过来，期望 1 个能成为“下一个的网景公司”（TNN）。如果其中的 7 个失败了也没什么，因为这笔交易经过精心设计，从 TNN 中赚的钱足够补偿所有损失。

风险投资家收到太多的商业计划，他们会拒绝 1000 份中的 999 份。他们的最大问题就是要过滤大量的商业计划，就像大海捞针一样找到他们认为值得投资的对象。他们相当擅长说“不”，但是不擅对坏的计划说“不”，对好的计划说“是”[②]。

寻求风险投资的原因包括以下几点：[③]

- 这家企业本身能因为被聪明的投资人投资而获益。
- 投资人能给公司拿出有价值的投资建议、关系和介绍。
- 企业有潜力成为一家大公司或成为一家大型上市公司。
- 初创企业的创立者并不拘泥于“他们的方式，或快速发展的道路”，并愿意放弃一些控制权使企业更成功。

不要寻求风险投资的原因包括：[④][⑤]

- 企业创立者厌恶风险，更愿意采用低风险的小额支付交易。
- 企业创办者只懂技术，没有商业经验，并希望永远绝对地控制企业。
- 投资者通常是“哑巴资金”，也就是说，这些人根本不了解这个领域——很有名的牙医乐于给你 50 万美元却不知道 CPM 和 CPC 和 CTO。
- 你打算进入一个既定的充满竞争的行业，一味贪快是没有好处的。你最好慢些开始，找到业务的新契机，从中得到发展，悄悄地把顾客从竞争者那边拉过来。
- 该产品还不成熟，未经证实，在这种情况下进行的昂贵营销努力都会白白浪费，它只能向世界证明你的产品有多么糟糕。

如果你成功得到风险投资，那么你的企业就会进入审核程序，或受到尽职调查。审核程序会根据投资的类型和规模而有所差别，但一般都会持续几个月。审核的部分程序是有关投资前和投资后的评估工作。这就决定了有多少该企业的份额（或产权）需要移交给风险投资公司以换取他们的投资。评估初创企业是一种主观的活动，创业家和投资者之间需要进行大量的协商。（在 Herb Cohen 的 *You Can Negotiate Anything* 这本书中，你可以找到很多的协商指南）[⑥]

---

① White paper by Mark A. Zawacki, “Startup Candy Vol. 1,” The Milestone Group, November 2009.

② Adapted from Michael Treacy and Fred Wiersema, The Discipline of Market Leaders (Addison-Wesley Publishing,1995).

③ Joel Spolsky, “Fixing Venture Capital,” June 03, 2003, http://www.joelonsoftware.com/articles/VC.html.

④ Ibid.

⑤ Joel Spolsky, “Raising Money for StackOverflow,” February 14, 2010, http://www.joelonsoftware.com/items/2010/02/14.html.

⑥ Herb Cohen, You Can Negotiate Anything (Bantam Press, 1982).

关键时刻获得资助

首先是互联网泡沫破裂，到 2000 年经济大萧条，然后是 2009 年的金融危机。尽管这种经济周期没什么么可新奇的，但是最近经济衰退的程度对需要资金的初创企业们更具挑战性。2009 年金融危机最严重的时候，常见一些风险投资家把已经产生利润和收入作为投资初创企业的条件。难怪风险投资几近枯竭。

所有这些教训就是让初创企业知道，要慎重管理现金流量，这样即使在困难时期也能成为合适的投资候选人。风险投资公司有句名言，缩写是 CIMITYM：Cash Is More Important Than Your Mother（现金比什么都重要）。这是因为一个企业没有母亲也能生存，但要是没有现金——情况就不一样了，这代表着风险投资家们的总体心态。

如果你恰巧在信用危机时期要寻求资助，那么就需要向潜在的投资人证明你的企业有创造并保持现金流的能力。具体来说，这就可以理解为把未来的支出与收益联系起来。通常的做法，举个例子来说就是展示你的 ROI（投资回报率）以及销售计划。这个可以用一个比率或百分比来表达。一个基本的计算公式如下：

ROI=（投资回报–投资）× 100/投资

比如你在销售活动中花了 3 万美元，你预计能产生 8 万美元的回报。那么 ROI 公式将是

ROI=（80 000–30 000）× 100/30 000=167%

计算初创企业投资前价值的最简单方法，就是把最近收到投资的情况相似的初创企业作为基准进行比较。投资后价值就是该初创企业在接受外部投资后的价值。属于投资者的份额就是他们购买的股票数量除以总流通股：

属于风险投资者的份额=风险投资者的股份 ÷ 投资后的总流通股

例如，如果你的投资前估值是 600 万美元，风投公司投资了 1200 万美元，投资后的估值就是 1800 万美元。风投公司在投资后将拥有该企业 2/3 或者 66.7%的份额（1200 万美元/1800 万美元=66.7%）。在投资前，如果你有 200 万的流通股，就能计算每股的价格：

股价=投资前估值 ÷ 投资前股份=600 万美元 ÷ 200 万美元=3 美元

你也可以计算股份数量：

股票发行量=投资额 ÷ 股价=1200 万美元 ÷ 3 美元=400 万美元

关键要记住的是用投资前的股份数量很容易计算股价，用投资后股份数量很容易计算所有权份额；通过增加或减少投资的数额来回切换。股票价格在投资前后是一样的，这一点也很重要。

通过尽职审查后，风险投资公司就会给你一个投资条件清单。该文件概述了将对公司投资的投资者条款。投资条款往往包括三个部分：资金、公司治理结构和清算。除了公司估值决定分配给风险投资公司的股本以外，风险投资公司将会对其投资定义一个新的止损策略。在附录 C 中你可以看到一个典型的条款清单实例。

我们现在已经知道了资金的四个主要来源，以及怎样为当前的公司发展阶段匹配这四种资助方式。接下来，我们将会了解怎样保护应用开发带来的宝贵资产：知识产权。

12.3　固守你的经营策略

尽管开发者出身的企业家不可能通晓所有法律条文，但知道一些基本原则是必需的。知识产权，或者 IP，是用来描述“创意产品”的术语。通过知识产权法可以保护的公司资产主要有两种：公司的品牌/标志设计，或者商标（延伸到移动/网络领域），以及与你发明的产品相关的知识产权（专利）。知道你能保护什么以及如何保护，就能确保你为开发做出的努力不会白费，并帮助你在此基础上发展业务。

三个遇到理想投资者的实用技巧

网络，网络，网络！

如果你是那种羞于向别人介绍自己的人，找一个不是这种类型的商业合作者！作为一个寻求资助的企业家，你需要把自己 30 秒钟的“电梯演讲”磨练到完美的程度。你可能需要一次又一次地使用这种电梯演讲，有可能完全出乎意料。你应该在哪里进行宣传？任何地方。什么时候？无时无刻。明白了？

时刻准备着，从而提高遇到理想投资者的机会。你可以加入企业家网络，商业俱乐部以及行业协会，并提交申请让你刚起步的企业加入特定的创业者竞赛。这里是一些你应该注意的网站，以便与潜在的投资者建立联系：

在线网络——这些有助于建立/维持相关联系的数据库。

- www.linkedin.com
- http://mobiforge.com/

有线下活动支撑的商业网络——这些对于联合同行的企业家、开发者和投资者非常重要。

- www.firsttuesday.com
- www.mobilemonday.net
- www.wipconnector.com

全球活动——参加或赢得初创企业竞赛，这些活动能促进刚起步的企业发展成为行业巨星。

- http://www.ctiawireless.com
- http://www.mobileworldcongress.com
- http://www.ted.com
- http://venturebeat.com/events/
- http://www.leweb.net/
- http://techcrunch.com/category/events/
- http://www.demo.com/

12.3.1 商标注册和品牌保护

商标注册从属于国家法律，因此要遵循的确切程序会有赖于你注册的地区和国家。在美国，美国专利与商标局负责管理联邦商标注册。按照《马德里议定书》，你可以申请国际商标保护。在欧盟国家，你可以注册一个社区商标（这是较为常见的），该商标能在全欧盟地区得到保护。

要申请注册商标，必须要记住可能有人会反对你的要求（例如，如果你尝试注册一个与现有的品牌相似的名称）。通常情况下，你的律师会进行初步检索，看看是否已经有一个竞争商标存在。在你提出申请之后，第三方需要一段时间（通常是 3 个月），在此期间，他们可以驳回你的请求。

正常的程序是你要选择哪个商业类的注册商标。这些国际公认的商品和贸易服务类别，是由世界知识产权组织定义的，可以分解为 45 个部分。能直接适用于软件开发的类别如下：

- 类 38——通信
- 类 42——科学技术服务和与之相关的研究与设计；工业分析和研究服务；电脑硬件及软件设计和研发。

如果你需要连带公司的商标与商品一起销售，就必须再注册其他类别。

通常，主要有四类商标：

- 文字商标——例如，有关公司或产品的一个词，或者一个自造的词或一个能让人联想到产品或服务的词。通常是指你的品牌名称。
- 图形商标——商标的视觉设计，例如，企业标志（用特殊形状的字创建一个图形）或一张图片，有特定形式的一个符号，一个标签，或字母或数字。
- 组合商标——文字和图形组合的商标。
- 外观商标——专门的包装设计或商品上一个独特的细节。

一个商标必须能区分你提供的商品和服务。可以把它看作一种标志，将你的商品或服务与其他人的区分开来。

如果有下列情况，就不能注册商标

- 描述了商品或服务或其任何特征的词，例如，展示了商品或服务的质量、数量、目的、价值或地理来源。
- 你所在行业的惯例。
- 无法区分。
- 三维的，但该形状仅由商品自身性质产生、是为获得技术效果而必须具有的形状，或者是使商品具有实质性价值的形状。
- 是特别保护的标志，比如王室标志
- 具有攻击性
- 违反法律，例如，宣传违禁药物
- 具有欺骗性。商标上应该没有任何误导公众认为你的商品和服务具有一种他们并没有的品质。

一般情况下，注册商标必须每 10 年续展一次以维持它的法律效力。

要记住很重要的一点是在美国任何时候申请商标注册，不管你是否在 USPTO（The United States Patent and Trademark Office，美国专利高标局）递交了申请，你都可以使用指定的 TM（注册商标）或者 SM（服务标志）提醒公众尊重你的权利。只有在 USPTO 真正注册之后，才能使用联邦注册标志®。

注册商标的有利方面如下：[①]

- 没有你的允许，他人不能使用你的商标。
- 它可以让你对任何未经许可就使用该商标的人提起法律诉讼。
- 如果有人使用你的商标，官方就能对冒用者提出起诉。
- 它是你的资产，你可以卖掉它或者批准他人使用。

经常被问到的商标保护的问题

如果我把企业标志注册为黑色和白色，是不是就意味着注册的商标使用所有颜色都是有效的，或者我是否需要指定我想用的颜色？

如果你注册了一个商标，你的注册权利与备案内容（例如，黑色和白色）一致。如果你申请的是特定的颜色组合，那么你的权利也限于该组合。

侵权范围会扩展至类似标志，因而保护也会扩展到其他颜色组合。具体情况，要看你的产品和服务，以及颜色可能带来的影响。

没有滴水不漏的规则，因为每个都要具体问题具体分析。

注册要用多长时间？

如果国家或国际商标审查员没有提出反对意见，而你的商标又不是对抗性的，通常在美国需要用大约 12 到 18 个月的时间完成注册，在欧洲可能要花费 6 个月的时间。如果有反对意见，或者如果你的商标是对抗性的，那么注册需要花费的时间就会更长。

要花多少钱？

在美国，注册一个多达三类的商标，自己做的话将花费大约 1500 美元，如果聘请律师来申请的话，花费很容易就会翻倍。在欧洲注册的能够在其 27 个欧盟成员国都有效的多达三类的社区商标，花费大概与在美国的一样。

12.3.2　为你的成果申请专利

专利法和商标法一样，各国之间互不相同。一些亚洲国家在这方面声名狼藉，他们执法特别松，这导致山寨货横行。美国的发明者可以为软件代码申请专利，而欧洲的专利法就不允许这样做。图 12-7 展示的是 Apple 公司在 2010 年 3 月为 iGroups 软件申请专利的文件，该软件可以让某些地方的人们在某些时候“聚到一起”。申请专利通常都要涉及大量的技术细节，还要提供大量详尽的示意图。

①英国专利局知识产权局, http://www.ipo.gov.uk/types/patent/p-about/p-whatis.htm。



图 12-7 苹果公司 2010 年 3 月申请的 iGroups 软件专利中的一张原理图，iGroups 运用 iPhone 手机定位技术广播和接收位置记号，这些记号都有时间戳，能在给定时间显示在给定位置出现的“人群”

在美国申请专利的优势在于手机应用开发者可以保护自己的发明创造。不利的方面就是申请专利的数量太过庞大，因此更需要原创性，就更需要广泛调查，以确保没有相似的专利已经在 USPTO 申请过。

如果你想在欧洲申请专利，那么专利律师可能会建议你申请一个软件和硬件相结合的专利。按其要求，硬件并不一定需要是新的发明，但是你需要证明将软硬件两者结合能确实产生新的功能。在实践中，这就意味着用户要想办法，或者使用案例，充分说明并证实该产品具有独特性。

什么是专利？

专利保护新的发明，包括其原理，能做什么，如何实现，由什么组成，如何制造。如果没有获得专利权人的同意，其他人无权制造、使用、引入或销售该发明成果。

要给你的发明成果申请一个新专利，该发明必须

- 有新颖性；
- 有独创性，对具有该领域知识和经验的人也不容易做出来；
- 有实用性。

取得专利也可以：

- 卖掉发明成果及所有的知识产权（IP）；
- 许可其他人使用该发明成果，但保留所有的知识产权。第 8 章我们已介绍过这是一种可能的创收策略；
- 与其他人讨论该发明，从而基于该发明开展业务。如果需要向某些投资者在尽职调查期间提供保密文件，有专利保护是很有用的。

与注册商标一样，申请专利首先要进行查询，看看以前是否存在相同的技术专利。可以进行两类搜索：

- ❑ 专利要件搜索——专利要件搜索可以找到与新奇事物以及产品发明或者过程有关的专利或其他出版物。
- ❑ 自由专利搜索——能发现你的想法是否会侵犯其他专利。

专利能给你什么样的保护？

申请专利的目的就是保护发明人的知识产权。在没有经过专利持有人许可的情况下，专利能禁止除了专利所有人之外的任何其他人制造或销售该专利的项目（或使用相应的业务方法）。

保护发明人的所有权是考虑申请专利的主要原因。如果要拥有发明所有权，尽快申请专利是很有必要的。

技术专利正成为科技公司的竞争优势。大公司保持其竞争优势的某些做法是存在争议的。除了标准，防御性的专利策略一般都是公司申请为了确保能被实际采用的，但攻击性专利策略却越来越多。攻击性专利策略就是为了构建阻止竞争者获得私有技术的壁垒。诺基亚和高通已被专利纠纷困扰多年。

你现在已经很清楚怎样通过商标和专利来保护知识产权了，也知道了专利技术是如何成为公司的关键竞争优势，同时也是大公司准备为之激烈斗争的。

常见的关于专利的问题

在获得专利之前，需要充分测试你的新成果吗？

不用。专利法规定，只要理论上能行得通，不需要证明该发明能工作。

获得研发成果专利需要多长时间？

这要看你选择的专利类型。如果你申请的是全球专利，遵守 Patent Cooperation Treaty (专利合作协定, PCT),那么完成该申请需要 30 个月。要取得美国的国家专利，得花费 3 年的时间。

申请专利需要花多少钱？

同样，这要看是哪种类型的专利。申请一个全球专利花费大约 8000 到 9000 美元。

怎样才能知道是否有类似的已经申请过的专利？

在美国，USPTO 能提供在线数据库，你可以在数据库中免费检索。在欧洲，门户网站 Espacenet 能提供同样的检索功能。

诺基亚与高通

过去几年，诺基亚与高通的法律争议一直是头号新闻。鉴于该争议涉及那么多的钱，广受关注也就不奇怪了：在 2006 年诺基亚由于使用高通的 CDMA 标准，向高通支付了 4.5 亿美元。随着新标准 WCDMA 的即将问世，诺基亚看到了一个可以减少其向高通支付专利费的机会。诺基亚指责高通非法复制其 6 个在软件应用移动下载和移动电视广播方面的专利。诺基亚声称高通对 WCDMA 标准的贡献要远远小于对 CDMA 标准的贡献。而高通声称诺基亚在美国的法律诉讼是“典型的一报还一报”，完全是为了干扰一直进行的专利费协商。

12.4 小结

尽管拟定商业计划看起来像是写摘要，但其作用则是支撑企业家向其利益相关人，主要是潜在的投资者兜售他们的想法。如果初创企业想要快速成长，那么可能早晚都需要寻求一些资金资助。有时，这就像是买彩票，中奖的概率很小。找到理想的投资者并把握能够表达你想法的最好机会，就可以提高成功的可能性。有了自己的想法和业务基础，就值得对它进行保护使其不受竞争者的影响，从而确保自己的品牌和软件的知识产权。尽管此过程很考验耐性，即使对适应性最强的企业家而言也是如此，但是为了确保商业成功，它值得你进行长期投资。

当心专利投机人

专利投机人是指在没有正当商业动机的前提下，提出侵略性法律诉讼的公司，其目的纯粹是为了谋取物质利益。专利投机人经常这样干：

- 从破产的公司购买专利，然后起诉另一家公司，声称其某个产品侵犯了它购买的某个专利。
- 在自身并没有意图制造该专利产品或提供专利服务的情况下，向声称的侵权人主张权利。
- 根本没有生产或研究条件，却强制主张专利。
- 尽管多数的初创企业能够免受专利投机人的影响（因为他们的目标是大公司），但这也说明要确保个人的专利足够有特色，要有详细说明，以避免将来遇到麻烦。

等等，还有几句话……

终于看完这本书了——希望你阅读愉快！正如前言中提到的，我们以独特的方式在本书中介绍了 LBS 的关键技术和商业应用。这样做是为了让读者开发出杀手级的位置感知应用，也因为我们意识到越来越多的开发人员希望成为创业者。本书因此还介绍了一些实用的商业知识，这对企业成功很有必要（谁知道呢，也许你们中的一位就是将来移动界的马克·扎克伯格！）。既然本书为你开发实用、好玩而又受欢迎的位置应用提供了基础，那剩下揭开自己移动开发新篇章的事儿就看你的了。

附　录 A

第 2 章中用到的 Java 示例代码

下面是利用 GPS 获取手机位置的 Java 代码。

代码清单 A-1　从内置 GPS 接收器中取得手机的位置

```
Criteria cr = new Criteria();
cr.setPreferredPowerConsumption(Criteria.POWER_USAGE_LOW
cr.setAddressInfoRequired(false);
cr.setAltitudeRequired(false);
cr.setSpeedAndCourseRequired(false);
cr.setHorizontalAccuracy(Criteria.NO_REQUIREMENT);
cr.setVerticalAccuracy(Criteria.NO_REQUIREMENT);

//cr.setPreferredResponseTime(30);
locationProvider = LocationProvider.getInstance(cr);
```

下面是利用辅助 GPS 获取手机位置，同时限制电量消耗的 Java 代码示例。

代码清单 A-2　从内置 AGPS 接收器取得手机的位置

```
cr.setPreferredPowerConsumption(Criteria.POWER_USAGE_LOW
cr.setCostAllowed(MIDlet.useAssistedGPS);
cr.setAddressInfoRequired(false);
cr.setAltitudeRequired(false);
cr.setSpeedAndCourseRequired(false);
cr.setHorizontalAccuracy(Criteria.NO_REQUIREMENT);
cr.setVerticalAccuracy(Criteria.NO_REQUIREMENT);

//cr.setPreferredResponseTime(30);
locationProvider = LocationProvider.getInstance(cr);
```

以下是 cell/get API 键返回用户确切位置的数据格式。

代码清单 A-3　cell/get API 键返回的数据格式如下

```
<rsp stat="ok">
<cell range="6000" lac="0" lat="53.4044704705058" nbSamples="47"
     lon="27.4110088888429" cellId="29513" mcc="250" mnc="99"/>
</rsp>
```

附 录

商业计划大纲

B

1. 封面
2. 概要
3. 目录
   A. 理念和商业机会分析
      1. 理念描述
      2. 为什么该理念代表好的商业机会?
      3 竞争策略
      4. 常规目标
   B. 企业家或开发团队描述
   C. 商业环境研究
      1. 影响项目的外部因素
      2. 影响市场的力量
      3. 风险、机遇和威胁分析
   D. 公司结构
   E. SWOT（优点、缺点、机遇和威胁）矩阵
   F. 研究和开发计划
   G. 协议、联盟以及外包
   H. 营销计划
      1.目标市场分析
      2. 营销组合
         - ❑ 产品
         - ❑ 地点
         - ❑ 定价
         - ❑ 促销
      3. 客户关系管理
   I. 质量管理
   J. 产品计划

K. 人力资源计划

L. 财务计划

1. 投资计划
2. 项目现金流和其他
3. 收支平衡分析

M. 启动程序

1. 先决条件
2. 公司流程
3. 筹资和/或资金来源

N. 应急计划

附 录 C

投资建议书（样本）

本建议书涉及创业公司 XYZ（以下简称“公司”），位于美国某地，以及投资人 ABC（以下简称“投资人”），位于某地（插入地址）。

公司

创业公司 XYZ

创建人

待定

待定

投资人

投资人 ABC

投资额

100 万美元

投资形式

股权：普通股（“新股”）

交易结构

创建人和投资人将共同持有公司股份。建议书中在提到“公司”时，适当的时候也适用于持股人，例如，任何股份转让或运营事项也应该适用于集团中的所有公司。最终的交易结构将根据税收和法律顾问的建议决定。

资金用途

公司将根据商业计划使用新资金为融资业务融资。

付款日期

按日期——100 万美元

如果公司需要进一步的外部资金支持，在公司和投资人对合适的报价达成共识时，投资人有可能对公司进一步投资。

投资前估值

完全摊薄股票及确定新股数量后，投资前估值是 1000 万美元。

原始股价格

每股新股票的原始购买价格等于投资前估值除以流通股和股票等价物，包括已授予的期权和

根据已有股份认购权及下面“股票期权计划”中规定可授予的期权。

分红

公司创建的前三年不分红。

估值调整及反摊薄条款

进一步融资或 IPO 的前期估值可以低于本轮融资的后期估值，投资人的实际转换价格也会相应调整，使得这一轮投资的认购投入对应的预期估值保持最低。

转换率

每份新股需要转换的普通股数量将由原始购买价格除以转换价格决定。最初的“转换价格”等于原始购买价格。转换价格将根据前面的 “反稀释条款”进行调整。

表决权

根据适用的法律，投资人将与其他公司的控股人（在 1:1 基础上）以股票数量为基础一起投票表决。

反摊薄/优先认购权

如果公司的股本增加（不含向员工/顾问发行的行使股票期权的股票），那么投资人将有权维持预期增长之前所持有的公司股份比例。如果投资人不希望增资，那他们有权（在本协议签订之日起两年内）获得初始投资的退款，以及投资额 6%的利息收入。

退出者

未来五年，任何以“良好退出者”（待定义）身份离开公司的创立者，他们的股份将由公司酌情回购，而留下的公司创立人按其他股东的现有持股比例分配这些股份。这种回购将按公平的市场价格回收。

任何以“不好的退出者”身份离开公司的创始人，他们一定比例的股票将由公司酌情按票面价值回购，而留下的人将这些股份按其他持股人的现有持股比例进行分配，分配比例为：

- 在 12 个月内离开 80%
- 在 12 个月到 24 个月之间离开 60%
- 在 24 个月到 36 个月之间离开 40%
- 在 36 个月到 48 个月之间离开 20%
- 在 48 个月之后离开 0%

要像创始人是“好的退出者”那样保持股份平衡。

优先权（转让）

如果股票持有者有意卖出他的全部或部分股份，其他股票持有人享有优先购买的权利。

套牢和互购

未经投资人事先书面同意，管理层不得销售、抵押或处置他们的股票。三年后，创始人遵循投资者第一优先购买权规定，总共可以转让其各自五分之一的股权。

如果受让人同意这种转让和互购限制，那么该条款不适用于向家庭成员、家族信托公司、房地产权利人，或者子公司机构投资者转让的情况。

遵从如前所述的优先购买权和互购权的当事人不会向投资竞争公司的竞争者或实体销售其

股权。

跟卖权

按上述套牢条款，现有的一个或多个股东打算向第三方销售股份和/或者表决权时，如果不能让其他持股人按比例，按第三方提供的相同条件转让的话，那么这些持股人就不能向第三方转让股权。

准许转让

投资人可以将他们的股权转让给予公司，无需触发跟卖权、优先购买权、否决权或其他限制条件。

股票期权计划和虚拟股票计划

投资人和公司将确定双方都认可的新股票期权计划和/或新虚拟股票计划，由董事会批准。未经投资人许可，不能增加 SOP 和/或 PSP 或任何额外期权计划。

董事会

最初，董事会由五名董事组成。投资者在董事会中的权利，与其持有的公司股份比例相当。最初，投资者会任命一名执行董事，也允许有一位观察员参加董事会。投资者代表出席董事会会议应该报销差旅费。

其他初始董事应按如下条件任命：

- 创始人提名 3 人；
- 投资者提名业内专家 1 人；
- 创始人提名的 1 或多名有行业经验的非执行董事。

董事会应该定期召开会议，至少每季度一次。

投资者有权指定董事会秘书。

董事委员会

董事会将组建一个由两名执行董事组成的执行委员会，这两名董事由投资人任命，执行委员会至少每月开会一次，以审核公司的运营情况和工作业绩。

限制性条款

公司上市前，应事先就如下决定咨询投资人：

1. 修改或改变新股票的权利、优先权或特权；
2. 设立或发行任何种类或系列与新股票具有同等或优先权利的股票或其他抵押物；
3. 公司优质资本重组；
4.增加或减少法定股本数量；
5. 声明或支付股息或分配现金、股票或其他资产，或赎回普通股票或现有普通股票；
6. 修改、变更或取消任何影响新股票持有者的公司章程；
7. 公司解散、清算或清盘，或停止所有或相当一部分公司业务；
8. 确定公司上市前 IPO 的实质性条款和完成条件；
9. 在公司的普通经营过程之外进行的实质资产的出售、转让或处置；
10. 批准包括资金消耗速度预测和战略计划的年度预算；

11. 增减董事会成员；

12. 与其他企业、公司或实体合并或重组；

如果公司有子公司（不论现在还是将来），那么这些限制条件也同样适用于这些附属公司。

知情权

投资者有权知道如下信息：

- 在每一会计年度终了之后的 60 天之内，收到经会计事务所审计的年度财务报表（包括资产负债表、收入和现金流量报表）。
- 在每个季度结束的 30 日内，收到有关收入、毛利率、现金流量、销售状况和产品开发的季度报告。
- 每季度现金消耗速度检查。
- 每个季度要向董事会提交前述两个报告，以便董事会表达他们的立场。
- 至少在该计划执行前 30 天提交年度战略计划和预算；投资者及其顾问应该在合适的时间收到通知，充分了解所有公司的账簿和记录，他们有权审查公司的决定，有权检查公司的财产，有权咨询管理公司，并遵守所有保密承诺。

声明和担保

创始人应向投资人提供类似交易的说明和担保，以及每个担保人各自和连带的债务情况，以免出现问题。

知识产权和发明

公司中的每一个创立者和雇员都有义务使所有的发明都受到专利或其他知识产权的保护。

竞业禁止

创立者们要保证都不直接或间接地互相竞争，或持有竞争对手或其子公司股票。在停止公司任职 12 个月内，不应招聘公司的雇员或活跃客户，不应当披露任何公司的保密信息。

签约条件

成交条件包括但不限于：

- 令人满意的技术、财务（会计和商业计划）、法律和人力资源严格评估完成
- 法律文件初审合格
- 没有可能对公司价值产生直接不利影响的经济和/或相关事实或情况
- 根据现行的会计责任
- 经 XYZ 投资委员会的最终批准
- 陈述和担保的准确性
- 对市场条件的满足度
- 修改公司章程，赋新股票权利和选择权

签约计划

签署条款清单：　　　　　　日期

终止：　　　　　　　　　　日期

起草

投资协议之外的交易文件将由投资者的法律顾问起草。

独占权

该公司应该并且应该确保其董事、雇员和顾问，在签订本建议书之后的 1 个月内专门与投资者进行协商并讨论新股票发行问题。没有投资者书面同意，不能向第三方申请、回复提议，也不能接受任何新融资提议。

提供信息

在上述提到的独占期间，公司应该按投资者的要求提供所有可用的公司信息。

适用法律

此建议书适用于美国法律，适用于美国司法管辖的任何地方。

接受/拒绝

在所有各方签署后，本建议书即可生效。如果在第一方签署一周内还有其他方没有签署，那么本建议书将被视为无效。

保密和新闻发布

本建议书的内容，以及投资者和公司目前正在讨论的投资项目，将被所有各方视为保密内容。没有事先征得投资者和公司的许可，不能在新闻发布会发布任何有关最终投资（或其他收入）的信息。

法律约束

除了这部分以及“独占性”、“适用法律”、“接受/拒绝”，以及“保密和新闻发布”部分之外，本建议书不具有法律约束力，对任何一方不负有任何义务。

上述条款将作为公司建议书的概要，其主旨未包括投资协议确定包含的所有条款和条件。本建议书的目的仅供讨论而已，不应被视为一个提议或购买承诺，或一个提议或销售保证，或在这里描述的抵押物。除了提供为了约束各方的“法律约束”之外，此建议书并不会制约任何一方，除非最终交易文件得到执行。

____

创业公司 XYZ

____

投资人

“本书帮你解开基于位置的移动服务之谜。”

——亚马逊读者评论

Location-Aware Applications

LBS 应用开发



随着智能手机和平板电脑等移动设备的日益普及，LBS（Location-Based Services，基于位置的服务，亦称位置感知应用）即将成为移动计算时代最重要的组成部分。现在，是时候关注并探究这个可能影响我们未来生活方式的新领域了。

本书由国际知名LBS技术专家及移动互联网风险投资家共同撰写，内容全面、资料翔实、示例丰富。全书从LBS应用的背景及现状讲起，围绕技术和商业两个主题，全景式地介绍了主流定位技术（GPS、小区识别码、Wi-Fi等）、地图API（MapQuest、Google、Bing、Yahoo等）、消费者应用类型（导航、社交、娱乐及增强现实等）、移动平台（iOS、Android及其他平台）以及服务端整合等，同时给出了相应的代码示例。针对LBS应用如何在移动互联网时代脱颖而出，作者凭借多年创业经验，深入剖析了LBS应用的变现模式、发行策略、知识产权保护以及隐私问题，并且给出了创业融资方面的建议。

作为国内第一本专门介绍LBS应用开发的图书，本书值得所有移动开发人员、互联网产品经理、市场营销人员以及对LBS感兴趣的所有人阅读参考。

本书内容

- LBS应用的历史及现状
- 各种定位技术
- 主流地图选项
- 移动开发平台
- 消费型LBS应用
- LBS应用的营利创收模式
- LBS应用的隐私问题
- 发行及宣传策略
- 创业及融资建议



Richard Ferraro 现任Golden Gekko公司（http://www.goldengekko.com/）销售副总裁及移动业务发展总监，该公司是全球领先的移动应用方案提供商。Richard此前是LBS创业公司GeoMe Mobile创始人，他拥有阿尔塔行政管理学院国际工商管理硕士学位。



Murat Aktihanoglu 风险投资公司ERA（Entrepreneurs Roundtable Accelerator，http://eranyc.com/）创始人，该公司拥有120名创业导师、10位运营合作伙伴，为互联网及移动创业公司提供免费的办公场所、服务和指导。Murat毕业于土耳其毕尔肯大学。



图灵社区：www.ituring.com.cn

新浪微博：@图灵教育　@图灵社区

反馈/投稿/推荐信箱：contact@turingbook.com

热线：(010)51095186转604

分类建议　计算机/程序设计/移动开发

人民邮电出版社网址：www.ptpress.com.cn



ISBN 978-7-115-29534-7

定价：59.00元

欢迎加入

图灵社区

最前沿的IT类电子书发售平台

电子出版的时代已经来临。在许多出版界同行还在犹豫彷徨的时候，图灵社区已经采取实际行动拥抱这个出版业巨变。作为国内第一家发售电子图书的IT类出版商，图灵社区目前为读者提供两种DRM-free的阅读体验：在线阅读和PDF。

相比纸质书，电子书具有许多明显的优势。它不仅发布快，更新容易，而且尽可能采用了彩色图片（即使有的书纸质版是黑白印刷的）。读者还可以方便地进行搜索、剪贴、复制和打印。

图灵社区进一步把传统出版流程与电子书出版业务紧密结合，目前已实现作译者网上交稿、编辑网上审稿、按章发布的电子出版模式。这种新的出版模式，我们称之为“敏捷出版”，它可以让读者以较快的速度了解到国外最新技术图书的内容，弥补以往翻译版技术书“出版即过时”的缺憾。同时，敏捷出版使得作、译、编、读的交流更为方便，可以提前消灭书稿中的错误，最大程度地保证图书出版的质量。

最方便的开放出版平台

图灵社区向读者开放在线写作功能，协助你实现自出版和开源出版的梦想。利用“合集”功能，你就能联合二三好友共同创作一部技术参考书，以免费或收费的形式提供给读者。（收费形式须经过图灵社区立项评审。）这极大地降低了出版的门槛。只要你有写作的意愿，图灵社区就能帮助你实现这个梦想。成熟的书稿，有机会入选出版计划，同时出版纸质书。

图灵社区引进出版的外文图书，都将在立项后马上在社区公布。如果你有意翻译哪本图书，欢迎你来社区申请。只要你通过试译的考验，即可签约成为图灵的译者。当然，要想成功地完成一本书的翻译工作，是需要有坚强的毅力的。

最直接的读者交流平台

在图灵社区，你可以十分方便地写作文章、提交勘误、发表评论，以各种方式与作译者、编辑人员和其他读者进行交流互动。提交勘误还能够获赠社区银子。

你可以积极参与社区经常开展的访谈、审读、评选等多种活动，赢取积分和银子，积累个人声望。

ituring.com.cn